

第4部：『魔方陣の最新研究』

第1章：原初魔方陣とひな形変換法の基礎的研究

第2節：原初四方陣とひな形変換法：撰田 寛二

次は、原初四方陣と四次の各種魔方陣とを対比しながら二者の解集合を結び付け、「ひな形変換法」を用いて、解の間に一対一対応の関係を発見したい。

1. 原初四方陣と補数対称型四方陣の関係

最初に補数対称型四方陣（単に「対称四方陣」とも言う）を取り上げる。下の表は「原初四方陣」と「補数対称型四方陣」の解の例を最初の24解に限って例示したものだ。

** 原初四方陣の解の例 **

1/	2/	3/	4/	5/	6/
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 3 4 9 10 11 12 5 6 7 8 13 14 15 16	1 2 5 6 3 4 7 8 9 10 13 14 11 12 15 16	1 2 5 6 9 10 13 14 3 4 7 8 11 12 15 16	1 2 9 10 3 4 11 12 5 6 13 14 7 8 15 16	1 2 9 10 5 6 13 14 3 4 11 12 7 8 15 16
7/	8/	9/	10/	11/	12/
1 3 2 4 5 7 6 8 9 11 10 12 13 15 14 16	1 3 2 4 9 11 10 12 5 7 6 8 13 15 14 16	1 3 5 7 2 4 6 8 9 11 13 15 10 12 14 16	1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16	1 3 9 11 2 4 10 12 5 7 13 15 6 8 14 16	1 3 9 11 5 7 13 15 2 4 10 12 6 8 14 16
13/	14/	15/	16/	17/	18/
1 5 2 6 3 7 4 8 9 13 10 14 11 15 12 16	1 5 2 6 9 13 10 14 3 7 4 8 11 15 12 16	1 5 3 7 2 6 4 8 9 13 11 15 10 14 12 16	1 5 3 7 9 13 11 15 2 6 4 8 10 14 12 16	1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16	1 5 9 13 3 7 11 15 2 6 10 14 4 8 12 16
19/	20/	21/	22/	23/	24/
1 9 2 10 3 11 4 12 5 13 6 14 7 15 8 16	1 9 2 10 5 13 6 14 3 11 4 12 7 15 8 16	1 9 3 11 2 10 4 12 5 13 7 15 6 14 8 16	1 9 3 11 5 13 7 15 2 10 4 12 6 14 8 16	1 9 5 13 2 10 6 14 3 11 7 15 4 12 8 16	1 9 5 13 3 11 7 15 2 10 6 14 4 12 8 16

.....

** 補数対称型四方陣の解の例 **

1/	2/	3/	4/	5/	6/
1 15 14 4 12 6 7 9 8 10 11 5 13 3 2 16	1 15 14 4 8 10 11 5 12 6 7 9 13 3 2 16	1 15 12 6 14 4 7 9 8 10 13 3 11 5 2 16	1 15 12 6 8 10 13 3 14 4 7 9 11 5 2 16	1 15 8 10 14 4 11 5 12 6 13 3 7 9 2 16	1 15 8 10 12 6 13 3 14 4 11 5 7 9 2 16
7/	8/	9/	10/	11/	12/
1 14 15 4 12 7 6 9 8 11 10 5 13 2 3 16	1 14 15 4 8 11 10 5 12 7 6 9 13 2 3 16	1 14 12 7 15 4 6 9 8 11 13 2 10 5 3 16	1 14 12 7 8 11 13 2 15 4 6 9 10 5 3 16	1 14 8 11 15 4 10 5 12 7 13 2 6 9 3 16	1 14 8 11 12 7 13 2 15 4 10 5 6 9 3 16
13/	14/	15/	16/	17/	18/
1 12 15 6 14 7 4 9 8 13 10 3 11 2 5 16	1 12 15 6 8 13 10 3 14 7 4 9 11 2 5 16	1 12 14 7 15 6 4 9 8 13 11 2 10 3 5 16	1 12 14 7 8 13 11 2 15 6 4 9 10 3 5 16	1 12 8 13 15 6 10 3 14 7 11 2 4 9 5 16	1 12 8 13 14 7 11 2 15 6 10 3 4 9 5 16

19/	20/	21/	22/	23/	24/
1 8 15 10	1 8 15 10	1 8 14 11	1 8 14 11	1 8 12 13	1 8 12 13
14 11 4 5	12 13 6 3	15 10 4 5	12 13 7 2	15 10 6 3	14 11 7 2
12 13 6 3	14 11 4 5	12 13 7 2	15 10 4 5	14 11 7 2	15 10 6 3
7 2 9 16	7 2 9 16	6 3 9 16	6 3 9 16	4 5 9 16	4 5 9 16
.					

原初四方陣は、第一基本形が「正方配列」そのものだ。その他の解は、基本形の内部を規則的に並べ替えた形をしている。対称四方陣の解は、既に何度も見て来たものだ。ただし、このリストでは、反転・回転形を整理する前の形の原始解 384 個の一部を掲げてある。

はたして、これら原初四方陣と対称四方陣の解集合の間に、どんな対応関係が見つかるであろうか。そのためにどのような「ひな形変換法」を考えたらよいだろうか。

「ひな形」は、目的方陣たる対称四方陣の解集合の中から 1 つ選ばれる。

試みに原初四方陣の第一基本形と対称四方陣の第一解とを並べて数値の異同を具体的に調べてみよう。下図のようになる。半数の 8 個の数値が同じであり、残りの 8 個を交換すればよいことがわかる。交換の規則も、どうやら簡単に書けそう。これなら、そのまま「ひな形」になるのではないか。ひな形変換法の規則を、下のリスト damt1() のように記述する。

* 第一基本形 *	異同表	* ひな形 *																																																
<table border="1" style="width: 100%; text-align: center;"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	<table border="1" style="width: 100%; text-align: center;"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	<table border="1" style="width: 100%; text-align: center;"> <tr><td>1</td><td>15</td><td>14</td><td>4</td></tr> <tr><td>12</td><td>6</td><td>7</td><td>9</td></tr> <tr><td>8</td><td>10</td><td>11</td><td>5</td></tr> <tr><td>13</td><td>3</td><td>2</td><td>16</td></tr> </table>	1	15	14	4	12	6	7	9	8	10	11	5	13	3	2	16
1	2	3	4																																															
5	6	7	8																																															
9	10	11	12																																															
13	14	15	16																																															
1	2	3	4																																															
5	6	7	8																																															
9	10	11	12																																															
13	14	15	16																																															
1	15	14	4																																															
12	6	7	9																																															
8	10	11	5																																															
13	3	2	16																																															
〔 原初四方陣 〕		〔 対称四方陣 〕																																																

```
/**/
/* 「ひな形変換法」 */
void damt1(){
    t[1]=p[1];   t[2]=p[15];   t[3]=p[14];   t[4]=p[4];
    t[5]=p[12];  t[6]=p[6];    t[7]=p[7];    t[8]=p[9];
    t[9]=p[8];   t[10]=p[10];  t[11]=p[11];  t[12]=p[5];
    t[13]=p[13]; t[14]=p[3];   t[15]=p[2];   t[16]=p[16];
}
/**/
```

この変換法を原初四方陣の 384 形に逐一施す。それで、はたして対称四方陣の解 384 個の集合を重複無く欠落無く再構成できるものであろうか。上のサンプルに試行した感じでは、どうやらうまく行きそうだが ...。

問題は、原初四方陣全数の構成法だ。384 個もの異形をどうやって描き出すか。

結局のところ、普通の魔方陣を構成する際の普通の手順を踏襲するしか無いようだ。では、その場合の「基本図」と基本条件を表す「定義式」をどう仮定すべきか。

これは、やはり「ひな形」に教わるしかあるまい。原初四方陣をいかに変形しようとも、変換した結果、目的方陣があくまで対称四方陣でなくては困る。そのために各行列の定和と対称型の補数定和式をあらためて要請する。ひな形から次のような「定義式」を引き出すことにする。そして、それを元に原初方陣を構成しよう。

実際のプログラムは、次のように書いた。

先ず最初に普通のやり方で補数対称型四方陣を構成し、リストに残す。次に、原初四方陣を構成し、それを逐一「ひな形変換」して目的方陣を再構成する。そして結果を比較する。

```

/** 'Prototype' Squares and Self-Complementary MS44 */
/** 'ProtoT44SC.c' built by Kanji Setsuda */
/** on Mar.8, 2003; Sep.13, 2006 */
/** Working on MacOSX & Xcode 2.2 */
/**/
#include <stdio.h>
/**/
short cnt, cnt1, cnt3;
short LSM, PSM;
short nm[17], uflg[17];
short dnm[17], lnm[385];
short dt[7][17];
short st[385][17];
/**/
/* Sub-Routines */
/**/
void deep01(void), deep02(void), deep03(void), deep04(void);
void deep05(void), deep06(void), deep07(void), deep08(void);
void ansrecord(void);
/**/
void stp01(void), stp02(void), stp03(void);
void stp04(void), stp05(void), stp06(void);
void stp07(void), stp08(void), stp09(void);
void ansprint(void), pr6ans(void);
void damt(void);
short findnmb(void);
void prlnm(void), damt(void);
/**/
/* Main Program */
int main(){
short n;
printf("\n ['Prototype Squares' 4x4 and");
printf(" Self-Complementary Magic Squares 4x4 Recomposed]\n");
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
LSM=34; PSM=17; cnt=0;
deep01(); /* Make the Solution List of Self-Complementary MS44 */
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
cnt1=0; cnt3=0;
stp01(); /* Make 'Prototype Squares' and Apply 'DAMT' */
printf(" [Count = %d/%d]\n",cnt,cnt1);
prlnm();
printf("\n [OK!]\n");
return 0;
}
/**/
/* Make the Solution List of Self-Complementary MS44 */
/** Basic Forms: **

  2  3  4 | 1 | 2 | 3 | 4 | 1  2  3          | 1 | 15 | 14 | 4 |
  -----+-----+-----+-----+-----+-----+-----+-----+-----+
  6  7  8 | 5 | 6 | 7 | 8 | 5  6  7          | 12 | 6 | 7 | 9 |
  -----+-----+-----+-----+-----+-----+-----+-----+
 10 11 12 | 9 |10 |11 |12 | 9 10 11          | 8 |10 |11 | 5 |
  -----+-----+-----+-----+-----+-----+-----+-----+
 14 15 16 |13 |14 |15 |16 |13 14 15          |13 | 3 | 2 |16 |
  -----+-----+-----+-----+-----+-----+-----+-----+

** Basic Conditions: **
n1+n2+n3+n4=C ...rw1; | n1+n5+n9+n13=C ...c1;
n5+n6+n7+n8=C ...rw2; | n2+n6+n10+n14=C ...c2;
n9+n10+n11+n12=C ...rw3; | n3+n7+n11+n15=C ...c3;
n13+n14+n15+n16=C ...rw4; | n4+n8+n12+n16=C ...c4;
** Self-complementary Conditions for MS44: **
n1+n16=n2+n15=n3+n14=n4+n13=n5+n12=n6+n11=n7+n10=n8+n9=CC ...scc
*/

```

```

/* Set N1 & n16=PSM-n1 */
void deep01(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[1]=a; nm[16]=b;
      uflg[a]=1; uflg[b]=1;
      deep02();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N2 & n15=PSM-n2 */
void deep02(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[2]=a; nm[15]=b;
      uflg[a]=1; uflg[b]=1;
      deep03();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N3 & n14 */
void deep03(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[3]=a; nm[14]=b;
      deep04();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n4=LSM-n1-n2-n3 & n13 */
void deep04(){
  short a,b;
  a=LSM-nm[1]-nm[2]-nm[3];
  if((0<a)&&(a<17)){
    b=LSM-nm[14]-nm[15]-nm[16];
    if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        uflg[a]=1; uflg[b]=1;
        nm[4]=a; nm[13]=b;
        deep05();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set N5 & n12=PSM-n5 */
void deep05(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[5]=a; nm[12]=b;
      uflg[a]=1; uflg[b]=1;
      deep06();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n9=LSM-n1-n5-n13 & n8 */
void deep06(){
  short a,b;
  a=LSM-nm[1]-nm[5]-nm[13];
  if((0<a)&&(a<17)){
    b=LSM-nm[4]-nm[12]-nm[16];
    if(a+b==PSM){

```

```

        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[9]=a; nm[8]=b;
            deep07();
            uflg[b]=0; uflg[a]=0;}}}
    }
    /* Set N6 & n11 */
    void deep07(){
        short a,b;
        for(a=1;a<17;a++){b=PSM-a;
            if((uflg[a]==0)&&(uflg[b]==0)){
                uflg[a]=1; uflg[b]=1;
                nm[6]=a; nm[11]=b;
                deep08();
                uflg[b]=0; uflg[a]=0;}
            }
        }
    /* Set n7=LSM-n5-n6-n8 & n10 */
    void deep08(){
        short a,b,c,d;
        a=LSM-nm[5]-nm[6]-nm[8];
        if((0<a)&&(a<17)){
            b=LSM-nm[9]-nm[11]-nm[12];
            if(a+b==PSM){
                c=LSM-nm[3]-nm[11]-nm[15];
                d=LSM-nm[2]-nm[6]-nm[14];
                if((a==c)&&(b==d)){
                    if((uflg[a]==0)&&(uflg[b]==0)){
                        nm[7]=a; nm[10]=b;
                        uflg[a]=1; uflg[b]=1;
                        ansrecord();
                        uflg[b]=0; uflg[a]=0;}}}}
        }
    /**/
    /* Record the Answers */
    void ansrecord(){
        short n;
        st[cnt][0]=cnt+1;
        for(n=1;n<17;n++){st[cnt][n]=nm[n];}
        cnt++;
    }
    /**/
    /* Make 'Prototype Squares' and Apply 'DAMT' */
    /** Basic Forms for Prototype Squares 4x4: **
    3 2 16 13 3 2 16 13 3 2
    15 14 4 | 1|15|14| 4 | 1 15 14 | 1 | 2 | 3 | 4 |
    6 7 9 |12| 6| 7| 9 |12 6 7 | 5 | 6 | 7 | 8 |
    10 11 5 | 8|10|11| 5 | 8 10 11 | 9 |10|11|12 |
    3 2 16 |13| 3| 2|16|13 3 2 |13|14|15|16|
    15 14 4 1 15 14 4 1 15 14
    ** Basic Conditions for Prototype Squares 4x4: **
    n1+n15+n14+n4=C ...rw1; | n1+n12+n8+n13=C ...c1;
    n12+n6+n7+n9=C ...rw2; | n15+n6+n10+n3=C ...c2;
    n8+n10+n11+n5=C ...rw3; | n14+n7+n11+n2=C ...c3;
    n13+n3+n2+n16=C ...rw4; | n4+n9+n5+n16=C ...c4;
    ** Self-complementary Conditions for Prototype Squares 4x4: **
    n1+n16=n15+n2=n14+n3=n4+n13=n12+n5=n6+n11=n7+n10=n9+n8=CC ...scc
    */
    /* Set N1 & n16=PSM-n1 */

```

```

void stp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
stp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-n15 */
void stp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
stp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N14 & n3 */
void stp03(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[14]=a; nm[3]=b;
stp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n4=LSM-n1-n15-n14 & n13 */
void stp04(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[14];
if((0<a)&&(a<17)){
b=LSM-nm[3]-nm[2]-nm[16];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[4]=a; nm[13]=b;
stp05();
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set N12 & n5=PSM-n12 */
void stp05(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[12]=a; nm[5]=b;
uflg[a]=1; uflg[b]=1;
stp06();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void stp06(){
short a,b;
a=LSM-nm[1]-nm[12]-nm[13];
if((0<a)&&(a<17)){
b=LSM-nm[4]-nm[5]-nm[16];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){

```

```

        uflg[a]=1; uflg[b]=1;
        nm[8]=a; nm[9]=b;
        stp07();
        uflg[b]=0; uflg[a]=0;}}}}
}
/* Set N6 & n11 */
void stp07(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[6]=a; nm[11]=b;
            stp08();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void stp08(){
    short a,b,c,d;
    a=LSM-nm[12]-nm[6]-nm[9];
    if((0<a)&&(a<17)){
        b=LSM-nm[8]-nm[11]-nm[5];
        if(a+b==PSM){
            c=LSM-nm[14]-nm[11]-nm[2];
            d=LSM-nm[15]-nm[6]-nm[3];
            if((a==c)&&(b==d)){
                if((uflg[a]==0)&&(uflg[b]==0)){
                    nm[7]=a; nm[10]=b;
                    uflg[a]=1; uflg[b]=1;
                    ansprint();
                    uflg[b]=0; uflg[a]=0;}}}}
}
/**/
/* Print the Solutions */
void ansprint(){
    short cn,n;
    cn=cnt3*2;
    for(n=1;n<17;n++){dt[cn][n]=nm[n];}
    cnt1++; dt[cn][0]=cnt1;
    damt();
    for(n=1;n<17;n++){dt[cn+1][n]=dnm[n];}
    n=findnmb(); dt[cn+1][0]=n; lnm[n]++;
    cnt3++; if(cnt3==3){pr6ans(); cnt3=0;}
}
/**/
/* Print 3 Pairs of Solutions */
void pr6ans(){
    short l,l4,m,n;
    printf("%10d/P:%3d/S%19d/P:%3d/S%19d/P:%3d/S\n",
        dt[0][0],dt[1][0],dt[2][0],dt[3][0],dt[4][0],dt[5][0]);
    for(l=0;l<4;l++){l4=l*4;
        for(m=0;m<6;m++){
            for(n=1;n<5;n++){printf("%3d",dt[m][l4+n]);}
            if(m<5){
                if(m%2==1){printf(" ");}else{printf(" ");}}
        }
        printf("\n");
    }
}
/**/
/* Transform Prototype into Self-Complementary Object */
void damt(void){
    dnm[1]=nm[1]; dnm[2]=nm[15]; dnm[3]=nm[14]; dnm[4]=nm[4];
    dnm[5]=nm[12]; dnm[6]=nm[6]; dnm[7]=nm[7]; dnm[8]=nm[9];
}

```

```

    dnm[9]=nm[8];   dnm[10]=nm[10]; dnm[11]=nm[11]; dnm[12]=nm[5];
    dnm[13]=nm[13]; dnm[14]=nm[3];  dnm[15]=nm[2];  dnm[16]=nm[16];
}
/*      Proto/      /Model
  +-----+       +-----+
  | 1| 2| 3| 4|   | 1|15|14| 4|
  +-----+       +-----+
  | 5| 6| 7| 8|   |12| 6| 7| 9|
  +-----+       +-----+
  | 9|10|11|12|   | 8|10|11| 5|
  +-----+       +-----+
  |13|14|15|16|   |13| 3| 2|16|
  +-----+       +-----+
*/
/**/
/* Find the Original Number */
short findnmb(){
  short mflg,cn,m,n;
  for(m=0;m<cnt;m++){
    mflg=0;
    for(n=1;n<17;n++){if(st[m][n]==dnm[n]){mflg++;}}
    if(mflg==16){cn=st[m][0]; break;}
  }
  return cn;
}
/**/
/* Print the Correspondence Monitor */
void prlnm(){
  short n;
  printf("\n** Monitor List of Correspondence between Old and New **\n");
  printf("  ??:%2d",lnm[0]);
  for(n=1;n<=cnt1;n++){
    if(n%32==1){printf("\n%4d:",n);}
    printf("%2d",lnm[n]);
  }
  printf("\n");
}
/**/

```

プログラムで良く見てほしいのは、原初方阵の作り方だ。stp01以降のプロセスを、deep01以降と比較しながら、丁寧に見ていただきたい。定義する変数名だけ変えているが、手順を全く同じにしたのにお気づきだろうか。「逆関数」を求めているのである。

早速にも実行結果を示そう。原初四方陣とその変換結果をカップリングして表示した。

そして、変換結果の対称四方陣全数について、古いリストと比較対照して検査し、解の対応関係を逐一モニターしてみたのが、最後の表である。

*** 原初四方陣およびひな形変換法によって作った対称四方陣の解のリスト (部分) ***

('n/P' が原初四方陣の通し番号 ; 'S/n' が対称四方陣の旧番号)

1/P: 1/S				2/P: 2/S				3/P: 3/S															
1	2	3	4	1	15	14	4	1	2	5	6	1	15	12	6								
5	6	7	8	12	6	7	9	9	10	11	12	8	10	11	5	3	4	7	8	14	4	7	9
9	10	11	12	8	10	11	5	5	6	7	8	12	6	7	9	9	10	13	14	8	10	13	3
13	14	15	16	13	3	2	16	13	14	15	16	13	3	2	16	11	12	15	16	11	5	2	16
4/P: 4/S				5/P: 5/S				6/P: 6/S															
1	2	5	6	1	15	12	6	1	2	9	10	1	15	8	10	1	2	9	10	1	15	8	10
9	10	13	14	8	10	13	3	3	4	11	12	14	4	11	5	5	6	13	14	12	6	13	3
3	4	7	8	14	4	7	9	5	6	13	14	12	6	13	3	3	4	11	12	14	4	11	5
11	12	15	16	11	5	2	16	7	8	15	16	7	9	2	16	7	8	15	16	7	9	2	16

<p>7/P: 7/S</p> <p>1 3 2 4 1 14 15 4 5 7 6 8 12 7 6 9 9 11 10 12 8 11 10 5 13 15 14 16 13 2 3 16</p> <p>10/P: 10/S</p> <p>1 3 5 7 1 14 12 7 9 11 13 15 8 11 13 2 2 4 6 8 15 4 6 9 10 12 14 16 10 5 3 16</p> <p>13/P: 13/S</p> <p>1 5 2 6 1 12 15 6 3 7 4 8 14 7 4 9 9 13 10 14 8 13 10 3 11 15 12 16 11 2 5 16</p> <p>16/P: 16/S</p> <p>1 5 3 7 1 12 14 7 9 13 11 15 8 13 11 2 2 6 4 8 15 6 4 9 10 14 12 16 10 3 5 16</p> <p>19/P: 19/S</p> <p>1 9 2 10 1 8 15 10 3 11 4 12 14 11 4 5 5 13 6 14 12 13 6 3 7 15 8 16 7 2 9 16</p> <p>22/P: 22/S</p> <p>1 9 3 11 1 8 14 11 5 13 7 15 12 13 7 2 2 10 4 12 15 10 4 5 6 14 8 16 6 3 9 16</p> <p>25/P: 25/S</p> <p>2 1 4 3 2 16 13 3 6 5 8 7 11 5 8 10 10 9 12 11 7 9 12 6 14 13 16 15 14 4 1 15</p> <p>28/P: 28/S</p> <p>2 1 6 5 2 16 11 5 10 9 14 13 7 9 14 4 4 3 8 7 13 3 8 10 12 11 16 15 12 6 1 15</p> <p>31/P: 31/S</p> <p>2 4 1 3 2 13 16 3 6 8 5 7 11 8 5 10 10 12 9 11 7 12 9 6 14 16 13 15 14 1 4 15</p> <p>34/P: 34/S</p> <p>2 4 6 8 2 13 11 8 10 12 14 16 7 12 14 1 1 3 5 7 16 3 5 10 9 11 13 15 9 6 4 15</p>	<p>8/P: 8/S</p> <p>1 3 2 4 1 14 15 4 9 11 10 12 8 11 10 5 5 7 6 8 12 7 6 9 13 15 14 16 13 2 3 16</p> <p>11/P: 11/S</p> <p>1 3 9 11 1 14 8 11 2 4 10 12 15 4 10 5 5 7 13 15 12 7 13 2 6 8 14 16 6 9 3 16</p> <p>14/P: 14/S</p> <p>1 5 2 6 1 12 15 6 9 13 10 14 8 13 10 3 3 7 4 8 14 7 4 9 11 15 12 16 11 2 5 16</p> <p>17/P: 17/S</p> <p>1 5 9 13 1 12 8 13 2 6 10 14 15 6 10 3 3 7 11 15 14 7 11 2 4 8 12 16 4 9 5 16</p> <p>20/P: 20/S</p> <p>1 9 2 10 1 8 15 10 5 13 6 14 12 13 6 3 3 11 4 12 14 11 4 5 7 15 8 16 7 2 9 16</p> <p>23/P: 23/S</p> <p>1 9 5 13 1 8 12 13 2 10 6 14 15 10 6 3 3 11 7 15 14 11 7 2 4 12 8 16 4 5 9 16</p> <p>26/P: 26/S</p> <p>2 1 4 3 2 16 13 3 10 9 12 11 7 9 12 6 6 5 8 7 11 5 8 10 14 13 16 15 14 4 1 15</p> <p>29/P: 29/S</p> <p>2 1 10 9 2 16 7 9 4 3 12 11 13 3 12 6 6 5 14 13 11 5 14 4 8 7 16 15 8 10 1 15</p> <p>32/P: 32/S</p> <p>2 4 1 3 2 13 16 3 10 12 9 11 7 12 9 6 6 8 5 7 11 8 5 10 14 16 13 15 14 1 4 15</p> <p>35/P: 35/S</p> <p>2 4 10 12 2 13 7 12 1 3 9 11 16 3 9 6 6 8 14 16 11 8 14 1 5 7 13 15 5 10 4 15</p>	<p>9/P: 9/S</p> <p>1 3 5 7 1 14 12 7 2 4 6 8 15 4 6 9 9 11 13 15 8 11 13 2 10 12 14 16 10 5 3 16</p> <p>12/P: 12/S</p> <p>1 3 9 11 1 14 8 11 5 7 13 15 12 7 13 2 2 4 10 12 15 4 10 5 6 8 14 16 6 9 3 16</p> <p>15/P: 15/S</p> <p>1 5 3 7 1 12 14 7 2 6 4 8 15 6 4 9 9 13 11 15 8 13 11 2 10 14 12 16 10 3 5 16</p> <p>18/P: 18/S</p> <p>1 5 9 13 1 12 8 13 3 7 11 15 14 7 11 2 2 6 10 14 15 6 10 3 4 8 12 16 4 9 5 16</p> <p>21/P: 21/S</p> <p>1 9 3 11 1 8 14 11 2 10 4 12 15 10 4 5 5 13 7 15 12 13 7 2 6 14 8 16 6 3 9 16</p> <p>24/P: 24/S</p> <p>1 9 5 13 1 8 12 13 3 11 7 15 14 11 7 2 2 10 6 14 15 10 6 3 4 12 8 16 4 5 9 16</p> <p>27/P: 27/S</p> <p>2 1 6 5 2 16 11 5 4 3 8 7 13 3 8 10 10 9 14 13 7 9 14 4 12 11 16 15 12 6 1 15</p> <p>30/P: 30/S</p> <p>2 1 10 9 2 16 7 9 6 5 14 13 11 5 14 4 4 3 12 11 13 3 12 6 8 7 16 15 8 10 1 15</p> <p>33/P: 33/S</p> <p>2 4 6 8 2 13 11 8 1 3 5 7 16 3 5 10 10 12 14 16 7 12 14 1 9 11 13 15 9 6 4 15</p> <p>36/P: 36/S</p> <p>2 4 10 12 2 13 7 12 6 8 14 16 11 8 14 1 1 3 9 11 16 3 9 6 5 7 13 15 5 10 4 15</p>
---	---	---

37/P: 37/S	38/P: 38/S	39/P: 39/S
2 6 1 5 2 11 16 5	2 6 1 5 2 11 16 5	2 6 4 8 2 11 13 8
4 8 3 7 13 8 3 10	10 14 9 13 7 14 9 4	1 5 3 7 16 5 3 10
10 14 9 13 7 14 9 4	4 8 3 7 13 8 3 10	10 14 12 16 7 14 12 1
12 16 11 15 12 1 6 15	12 16 11 15 12 1 6 15	9 13 11 15 9 4 6 15
40/P: 40/S	41/P: 41/S	42/P: 42/S
2 6 4 8 2 11 13 8	2 6 10 14 2 11 7 14	2 6 10 14 2 11 7 14
10 14 12 16 7 14 12 1	1 5 9 13 16 5 9 4	4 8 12 16 13 8 12 1
1 5 3 7 16 5 3 10	4 8 12 16 13 8 12 1	1 5 9 13 16 5 9 4
9 13 11 15 9 4 6 15	3 7 11 15 3 10 6 15	3 7 11 15 3 10 6 15
43/P: 43/S	44/P: 44/S	45/P: 45/S
2 10 1 9 2 7 16 9	2 10 1 9 2 7 16 9	2 10 4 12 2 7 13 12
4 12 3 11 13 12 3 6	6 14 5 13 11 14 5 4	1 9 3 11 16 9 3 6
6 14 5 13 11 14 5 4	4 12 3 11 13 12 3 6	6 14 8 16 11 14 8 1
8 16 7 15 8 1 10 15	8 16 7 15 8 1 10 15	5 13 7 15 5 4 10 15
46/P: 46/S	47/P: 47/S	48/P: 48/S
2 10 4 12 2 7 13 12	2 10 6 14 2 7 11 14	2 10 6 14 2 7 11 14
6 14 8 16 11 14 8 1	1 9 5 13 16 9 5 4	4 12 8 16 13 12 8 1
1 9 3 11 16 9 3 6	4 12 8 16 13 12 8 1	1 9 5 13 16 9 5 4
5 13 7 15 5 4 10 15	3 11 7 15 3 6 10 15	3 11 7 15 3 6 10 15
49/P: 49/S	50/P: 50/S	51/P: 51/S
3 1 4 2 3 16 13 2	3 1 4 2 3 16 13 2	3 1 7 5 3 16 10 5
7 5 8 6 10 5 8 11	11 9 12 10 6 9 12 7	4 2 8 6 13 2 8 11
11 9 12 10 6 9 12 7	7 5 8 6 10 5 8 11	11 9 15 13 6 9 15 4
15 13 16 14 15 4 1 14	15 13 16 14 15 4 1 14	12 10 16 14 12 7 1 14
70/P: 70/S	71/P: 71/S	72/P: 72/S
3 11 4 12 3 6 13 12	3 11 7 15 3 6 10 15	3 11 7 15 3 6 10 15
7 15 8 16 10 15 8 1	1 9 5 13 16 9 5 4	4 12 8 16 13 12 8 1
1 9 2 10 16 9 2 7	4 12 8 16 13 12 8 1	1 9 5 13 16 9 5 4
5 13 6 14 5 4 11 14	2 10 6 14 2 7 11 14	2 10 6 14 2 7 11 14
73/P: 73/S	74/P: 74/S	75/P: 75/S
4 2 3 1 4 15 14 1	4 2 3 1 4 15 14 1	4 2 8 6 4 15 9 6
8 6 7 5 9 6 7 12	12 10 11 9 5 10 11 8	3 1 7 5 14 1 7 12
12 10 11 9 5 10 11 8	8 6 7 5 9 6 7 12	12 10 16 14 5 10 16 3
16 14 15 13 16 3 2 13	16 14 15 13 16 3 2 13	11 9 15 13 11 8 2 13
76/P: 76/S	77/P: 77/S	78/P: 78/S
4 2 8 6 4 15 9 6	4 2 12 10 4 15 5 10	4 2 12 10 4 15 5 10
12 10 16 14 5 10 16 3	3 1 11 9 14 1 11 8	8 6 16 14 9 6 16 3
3 1 7 5 14 1 7 12	8 6 16 14 9 6 16 3	3 1 11 9 14 1 11 8
11 9 15 13 11 8 2 13	7 5 15 13 7 12 2 13	7 5 15 13 7 12 2 13
190/P: 190/S	191/P: 191/S	192/P: 192/S
8 16 6 14 8 1 11 14	8 16 7 15 8 1 10 15	8 16 7 15 8 1 10 15
7 15 5 13 10 15 5 4	4 12 3 11 13 12 3 6	6 14 5 13 11 14 5 4
4 12 2 10 13 12 2 7	6 14 5 13 11 14 5 4	4 12 3 11 13 12 3 6
3 11 1 9 3 6 16 9	2 10 1 9 2 7 16 9	2 10 1 9 2 7 16 9
193/P: 193/S	194/P: 194/S	195/P: 195/S
9 1 10 2 9 16 7 2	9 1 10 2 9 16 7 2	9 1 11 3 9 16 6 3
11 3 12 4 6 3 12 13	13 5 14 6 4 5 14 11	10 2 12 4 7 2 12 13
13 5 14 6 4 5 14 11	11 3 12 4 6 3 12 13	13 5 15 7 4 5 15 10
15 7 16 8 15 10 1 8	15 7 16 8 15 10 1 8	14 6 16 8 14 11 1 8

2. 原初四方陣と完全四方陣の関係

つぎは、汎四方陣の場合を調べよう。

四次の魔方陣では、汎対角線型は、「完全型」でもあり「正方連結型」でもある。最初の定義を違えて別個に設計・構成しても、最終的に解集合は全部同じものが出る（いわゆる「奇跡の一致」だ。七不思議の一つ）。

今回は、完全型汎四方陣の定義を採用する。

完全型は、17の補数ペアが、全て汎対角線上に並ぶ。2組ずつ現れて、定和の34を実現する。それを定義条件に採用して、設計することにしよう。一度に2つの変数をまとめて決定できるので、プログラムの長さが半分になり、実行スピードも倍になる。

* 完全型汎四方陣の基本図と定義式 *

<table border="1" style="border-collapse: collapse; text-align: center; width: 100%;"> <tr><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>11</td><td>12</td><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>15</td><td>16</td><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>	3	4	1	2	3	4	7	8	5	6	7	8	11	12	9	10	11	12	15	16	13	14	15	16	<table border="0" style="width: 100%;"> <tr><td>1</td><td>2</td></tr> <tr><td>5</td><td>6</td></tr> <tr><td>9</td><td>10</td></tr> <tr><td>13</td><td>14</td></tr> </table>	1	2	5	6	9	10	13	14	<p>[基本条件式]</p> $n1+n2+n3+n4=C \quad \dots \text{rw1};$ $n5+n6+n7+n8=C \quad \dots \text{rw2};$ $n9+n10+n11+n12=C \quad \dots \text{rw3};$ $n13+n14+n15+n16=C \quad \dots \text{rw4};$ $n1+n5+n9+n13=C \quad \dots \text{cl1};$ $n2+n6+n10+n14=C \quad \dots \text{cl2};$ $n3+n7+n11+n15=C \quad \dots \text{cl3};$ $n4+n8+n12+n16=C \quad \dots \text{cl4};$	<p>[完全型汎四方陣の補数定和式]</p> $n1+n11=CC \quad \dots \text{Cmp1};$ $n2+n12=CC \quad \dots \text{Cmp2};$ $n3+n9=CC \quad \dots \text{Cmp3};$ $n4+n10=C \quad \dots \text{Cmp4};$ $n5+n15=CC \quad \dots \text{Cmp5};$ $n6+n16=CC \quad \dots \text{Cmp6};$ $n7+n13=CC \quad \dots \text{Cmp7};$ $n8+n14=CC \quad \dots \text{Cmp8};$
3	4	1	2	3	4																														
7	8	5	6	7	8																														
11	12	9	10	11	12																														
15	16	13	14	15	16																														
1	2																																		
5	6																																		
9	10																																		
13	14																																		

** 完全型汎四方陣の標準解 48 個のリスト (全) **

1/ 1 15 4 14	2/ 1 15 4 14	3/ 1 15 6 12	4/ 1 15 6 12	5/ 1 15 10 8	6/ 1 15 10 8
12 6 9 7	8 10 5 11	14 4 9 7	8 10 3 13	14 4 5 11	12 6 3 13
13 3 16 2	13 3 16 2	11 5 16 2	11 5 16 2	7 9 16 2	7 9 16 2
8 10 5 11	12 6 9 7	8 10 3 13	14 4 9 7	12 6 3 13	14 4 5 11
7/ 1 14 4 15	8/ 1 14 4 15	9/ 1 14 7 12	10/ 1 14 11 8	11/ 1 12 6 15	12/ 1 12 7 14
12 7 9 6	8 11 5 10	8 11 2 13	12 7 2 13	8 13 3 10	8 13 2 11
13 2 16 3	13 2 16 3	10 5 16 3	6 9 16 3	11 2 16 5	10 3 16 5
8 11 5 10	12 7 9 6	15 4 9 6	15 4 5 10	14 7 9 4	15 6 9 4
13/ 2 16 3 13	14/ 2 16 3 13	15/ 2 16 5 11	16/ 2 16 5 11	17/ 2 16 9 7	18/ 2 16 9 7
11 5 10 8	7 9 6 12	13 3 10 8	7 9 4 14	13 3 6 12	11 5 4 14
14 4 15 1	14 4 15 1	12 6 15 1	12 6 15 1	8 10 15 1	8 10 15 1
7 9 6 12	11 5 10 8	7 9 4 14	13 3 10 8	11 5 4 14	13 3 6 12
19/ 2 13 3 16	20/ 2 13 3 16	21/ 2 13 8 11	22/ 2 13 12 7	23/ 2 11 5 16	24/ 2 11 8 13
11 8 10 5	7 12 6 9	7 12 1 14	11 8 1 14	7 14 4 9	7 14 1 12
14 1 15 4	14 1 15 4	9 6 15 4	5 10 15 4	12 1 15 6	9 4 15 6
7 12 6 9	11 8 10 5	16 3 10 5	16 3 6 9	13 8 10 3	16 5 10 3
25/ 3 16 2 13	26/ 3 16 2 13	27/ 3 16 5 10	28/ 3 16 5 10	29/ 3 16 9 6	30/ 3 16 9 6
10 5 11 8	6 9 7 12	13 2 11 8	6 9 4 15	13 2 7 12	10 5 4 15
15 4 14 1	15 4 14 1	12 7 14 1	12 7 14 1	8 11 14 1	8 11 14 1
6 9 7 12	10 5 11 8	6 9 4 15	13 2 11 8	10 5 4 15	13 2 7 12
31/ 3 13 2 16	32/ 3 13 2 16	33/ 3 13 8 10	34/ 3 13 12 6	35/ 4 15 1 14	36/ 4 15 1 14
10 8 11 5	6 12 7 9	6 12 1 15	10 8 1 15	9 6 12 7	5 10 8 11
15 1 14 4	15 1 14 4	9 7 14 4	5 11 14 4	16 3 13 2	16 3 13 2
6 12 7 9	10 8 11 5	16 2 11 5	16 2 7 9	5 10 8 11	9 6 12 7

37/ 4 15 6 9 14 1 12 7 11 8 13 2 5 10 3 16	38/ 4 15 6 9 5 10 3 16 11 8 13 2 14 1 12 7	39/ 4 15 10 5 14 1 8 11 7 12 13 2 9 6 3 16	40/ 4 15 10 5 9 6 3 16 7 12 13 2 14 1 8 11	41/ 4 14 1 15 9 7 12 6 16 2 13 3 5 11 8 10	42/ 4 14 1 15 5 11 8 10 16 2 13 3 9 7 12 6
43/ 4 14 7 9 5 11 2 16 10 8 13 3 15 1 12 6	44/ 4 14 11 5 9 7 2 16 6 12 13 3 15 1 8 10	45/ 5 16 2 11 4 9 7 14 15 6 12 1 10 3 13 8	46/ 5 16 3 10 4 9 6 15 14 7 12 1 11 2 13 8	47/ 6 15 1 12 3 10 8 13 16 5 11 2 9 4 14 7	48/ 6 15 4 9 3 10 5 16 13 8 11 2 12 1 14 7

[Count = 48]

かかる目的方陣を，原初四方陣とひな形変換法だけで作ることができるものだろうか。

先ず上のリストの中から「ひな形」を選ばなければならない。暫定的に第1番解を「ひな形」に指名する。そして，次のようなプログラムを書いて，原初四方陣を構成する。

```

/**/
/* Make 'Prototype Squares' of Order 4: */
/** Basic Forms of Prototype Squares for Complete Pan-magic Type **
10 5 11 8 10 5 11 8 10 5
15 4 14 | 1|15| 4|14| 1 15 4      | 1| 2| 3| 4|
6 9 7 |12| 6| 9| 7|12 6 9      | 5| 6| 7| 8|
3 16 2|13| 3|16| 2|13 3 16      | 9|10|11|12|
10 5 11| 8|10| 5|11| 8 10 5      |13|14|15|16|
15 4 14 1 15 4 14 1 15 4
** Basic Conditions for Prototype Squares 4x4: **
n1+n15+n4+n14=C ...rw1; | n1+n12+n13+n8=C ...c11;
n12+n6+n9+n7=C ...rw2; | n15+n6+n3+n10=C ...c12;
n13+n3+n16+n2=C ...rw3; | n4+n9+n16+n5=C ...c13;
n8+n10+n5+n11=C ...rw4; | n14+n7+n2+n11=C ...c14;
** Complementary Conditions for Prototype Squares 4x4: **
n1+n16=n15+n2=n4+n13=n14+n3=n12+n5=n6+n1=n9+n81=n7+n10=CC ...cc
*/
/* Set N1 & n16=PSM-N1 */
void cstp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
cstp02();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set N15 & n2=PSM-N15 */
void cstp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
cstp03();
uflg[b]=0; uflg[a]=0;}}
}
}

```

```

}
/* Set N4 & n13=PSM-N4 */
void cstp03(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[4]=a; nm[13]=b;
      cstp04();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n14=LSM-n1-n15-n4 & n3 */
void cstp04(){
  short a,b;
  a=LSM-nm[1]-nm[15]-nm[4];
  if((0<a)&&(a<17)){
    b=LSM-nm[16]-nm[2]-nm[13];
    if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        uflg[a]=1; uflg[b]=1;
        nm[14]=a; nm[3]=b;
        cstp05();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set N12 & n5=PSM-N12 */
void cstp05(){
  short a,b;
  for(a=16;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[12]=a; nm[5]=b;
      uflg[a]=1; uflg[b]=1;
      cstp06();
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set N6 & n11=PSM-N6 */
void cstp06(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      uflg[a]=1; uflg[b]=1;
      nm[6]=a; nm[11]=b;
      if(nm[1]+nm[6]+nm[16]+nm[11]==LSM){cstp07();}
      uflg[b]=0; uflg[a]=0;}
  }
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void cstp07(){
  short a,b;
  a=LSM-nm[1]-nm[12]-nm[13];
  if((0<a)&&(a<17)){
    b=LSM-nm[16]-nm[5]-nm[4];
    if(a+b==PSM){
      if((uflg[a]==0)&&(uflg[b]==0)){
        uflg[a]=1; uflg[b]=1;
        nm[8]=a; nm[9]=b;
        if(nm[14]+nm[9]+nm[3]+nm[8]==LSM){cstp08();}
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void cstp08(){
  short a,b,c,d;
  a=LSM-nm[12]-nm[6]-nm[9];

```

```

if((0<a)&&(a<17)){
  b=LSM-nm[5]-nm[11]-nm[8];
  if(a+b==PSM){
    c=LSM-nm[14]-nm[2]-nm[11];
    d=LSM-nm[3]-nm[15]-nm[6];
    if((a==c)&&(b==d)){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[7]=a; nm[10]=b;
        uflg[a]=1; uflg[b]=1;
        reccpspr();
        uflg[b]=0; uflg[a]=0;}}}}
}
/**/
/* Record the Solutions */
void reccpspr(){
  short n;
  for(n=1;n<17;n++){d[cnt3][n]=nm[n]; ptc[cnt2][n]=nm[n];}
  ptc[cnt2][0]=cnt2+1;
  cnt2++; d[cnt3][0]=cnt2;
}
/**/
/* Print 3 Pairs of Solutions */
void pr3pairs(short x){
  short l,l4,m,n,p;
  for(m=0;m<x;m=m+3){
    printf("%10d/S%4d/C%19d/S%4d/C%19d/S%4d/C\n",
      pts[m][0],ptc[m][0],pts[m+1][0],ptc[m+1][0],pts[m+2][0],ptc[m+2][0]);
    for(l=0;l<4;l++){l4=l*4;
      for(n=m;n<(m+3);n++){
        for(p=1;p<5;p++){printf("%3d",pts[n][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",ptc[n][l4+p]);}
        if(n<(m+2)){printf(" ");}
      }
      printf("\n");
    }
  }
}
/**/

```

原初四方陣の構成の仕方は、補数対称型の時と全く同じ。「逆関数」を解く要領で実際に構成してみると、384 個の原初四方陣が得られた。何と、補数対称型の場合と全く同じ解集合であった。条件式が酷似しているためだろうか、そういう結果になった。

**** 補数対称型/S と完全型汎四方陣/C のための2つのタイプの原初四方陣の比較: ****

1/S				1/C				2/S				2/C				3/S				3/C			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	5	6	1	2	5	6
5	6	7	8	5	6	7	8	9	10	11	12	9	10	11	12	3	4	7	8	3	4	7	8
9	10	11	12	9	10	11	12	5	6	7	8	5	6	7	8	9	10	13	14	9	10	13	14
13	14	15	16	13	14	15	16	13	14	15	16	13	14	15	16	11	12	15	16	11	12	15	16
4/S				4/C				5/S				5/C				6/S				6/C			
1	2	5	6	1	2	5	6	1	2	9	10	1	2	9	10	1	2	9	10	1	2	9	10
9	10	13	14	9	10	13	14	3	4	11	12	3	4	11	12	5	6	13	14	5	6	13	14
3	4	7	8	3	4	7	8	5	6	13	14	5	6	13	14	3	4	11	12	3	4	11	12
11	12	15	16	11	12	15	16	7	8	15	16	7	8	15	16	7	8	15	16	7	8	15	16
7/S				7/C				8/S				8/C				9/S				9/C			
1	3	2	4	1	3	2	4	1	3	2	4	1	3	2	4	1	3	5	7	1	3	5	7
5	7	6	8	5	7	6	8	9	11	10	12	9	11	10	12	2	4	6	8	2	4	6	8
9	11	10	12	9	11	10	12	5	7	6	8	5	7	6	8	9	11	13	15	9	11	13	15
13	15	14	16	13	15	14	16	13	15	14	16	13	15	14	16	10	12	14	16	10	12	14	16

10/S	10/C	11/S	11/C	12/S	12/C
1 3 5 7	1 3 5 7	1 3 9 11	1 3 9 11	1 3 9 11	1 3 9 11
9 11 13 15	9 11 13 15	2 4 10 12	2 4 10 12	5 7 13 15	5 7 13 15
2 4 6 8	2 4 6 8	5 7 13 15	5 7 13 15	2 4 10 12	2 4 10 12
10 12 14 16	10 12 14 16	6 8 14 16	6 8 14 16	6 8 14 16	6 8 14 16
13/S	13/C	14/S	14/C	15/S	15/C
1 5 2 6	1 5 2 6	1 5 2 6	1 5 2 6	1 5 3 7	1 5 3 7
3 7 4 8	3 7 4 8	9 13 10 14	9 13 10 14	2 6 4 8	2 6 4 8
9 13 10 14	9 13 10 14	3 7 4 8	3 7 4 8	9 13 11 15	9 13 11 15
11 15 12 16	11 15 12 16	11 15 12 16	11 15 12 16	10 14 12 16	10 14 12 16
16/S	16/C	17/S	17/C	18/S	18/C
1 5 3 7	1 5 3 7	1 5 9 13	1 5 9 13	1 5 9 13	1 5 9 13
9 13 11 15	9 13 11 15	2 6 10 14	2 6 10 14	3 7 11 15	3 7 11 15
2 6 4 8	2 6 4 8	3 7 11 15	3 7 11 15	2 6 10 14	2 6 10 14
10 14 12 16	10 14 12 16	4 8 12 16	4 8 12 16	4 8 12 16	4 8 12 16
19/S	19/C	20/S	20/C	21/S	21/C
1 9 2 10	1 9 2 10	1 9 2 10	1 9 2 10	1 9 3 11	1 9 3 11
3 11 4 12	3 11 4 12	5 13 6 14	5 13 6 14	2 10 4 12	2 10 4 12
5 13 6 14	5 13 6 14	3 11 4 12	3 11 4 12	5 13 7 15	5 13 7 15
7 15 8 16	7 15 8 16	7 15 8 16	7 15 8 16	6 14 8 16	6 14 8 16
22/S	22/C	23/S	23/C	24/S	24/C
1 9 3 11	1 9 3 11	1 9 5 13	1 9 5 13	1 9 5 13	1 9 5 13
5 13 7 15	5 13 7 15	2 10 6 14	2 10 6 14	3 11 7 15	3 11 7 15
2 10 4 12	2 10 4 12	3 11 7 15	3 11 7 15	2 10 6 14	2 10 6 14
6 14 8 16	6 14 8 16	4 12 8 16	4 12 8 16	4 12 8 16	4 12 8 16
25/S	25/C	26/S	26/C	27/S	27/C
2 1 4 3	2 1 4 3	2 1 4 3	2 1 4 3	2 1 6 5	2 1 6 5
6 5 8 7	6 5 8 7	10 9 12 11	10 9 12 11	4 3 8 7	4 3 8 7
10 9 12 11	10 9 12 11	6 5 8 7	6 5 8 7	10 9 14 13	10 9 14 13
14 13 16 15	14 13 16 15	14 13 16 15	14 13 16 15	12 11 16 15	12 11 16 15
28/S	28/C	29/S	29/C	30/S	30/C
2 1 6 5	2 1 6 5	2 1 10 9	2 1 10 9	2 1 10 9	2 1 10 9
10 9 14 13	10 9 14 13	4 3 12 11	4 3 12 11	6 5 14 13	6 5 14 13
4 3 8 7	4 3 8 7	6 5 14 13	6 5 14 13	4 3 12 11	4 3 12 11
12 11 16 15	12 11 16 15	8 7 16 15	8 7 16 15	8 7 16 15	8 7 16 15
31/S	31/C	32/S	32/C	33/S	33/C
2 4 1 3	2 4 1 3	2 4 1 3	2 4 1 3	2 4 6 8	2 4 6 8
6 8 5 7	6 8 5 7	10 12 9 11	10 12 9 11	1 3 5 7	1 3 5 7
10 12 9 11	10 12 9 11	6 8 5 7	6 8 5 7	10 12 14 16	10 12 14 16
14 16 13 15	14 16 13 15	14 16 13 15	14 16 13 15	9 11 13 15	9 11 13 15
34/S	34/C	35/S	35/C	36/S	36/C
2 4 6 8	2 4 6 8	2 4 10 12	2 4 10 12	2 4 10 12	2 4 10 12
10 12 14 16	10 12 14 16	1 3 9 11	1 3 9 11	6 8 14 16	6 8 14 16
1 3 5 7	1 3 5 7	6 8 14 16	6 8 14 16	1 3 9 11	1 3 9 11
9 11 13 15	9 11 13 15	5 7 13 15	5 7 13 15	5 7 13 15	5 7 13 15
37/S	37/C	38/S	38/C	39/S	39/C
2 6 1 5	2 6 1 5	2 6 1 5	2 6 1 5	2 6 4 8	2 6 4 8
4 8 3 7	4 8 3 7	10 14 9 13	10 14 9 13	1 5 3 7	1 5 3 7
10 14 9 13	10 14 9 13	4 8 3 7	4 8 3 7	10 14 12 16	10 14 12 16
12 16 11 15	12 16 11 15	12 16 11 15	12 16 11 15	9 13 11 15	9 13 11 15
40/S	40/C	41/S	41/C	42/S	42/C
2 6 4 8	2 6 4 8	2 6 10 14	2 6 10 14	2 6 10 14	2 6 10 14


```

10 14 12 16 10 14 12 16 1 5 9 13 1 5 9 13 4 8 12 16 4 8 12 16
1 5 3 7 1 5 3 7 4 8 12 16 4 8 12 16 1 5 9 13 1 5 9 13
9 13 11 15 9 13 11 15 3 7 11 15 3 7 11 15 3 7 11 15 3 7 11 15
      43/S 43/C          44/S 44/C          45/S 45/C
2 10 1 9 2 10 1 9 2 10 1 9 2 10 1 9 2 10 4 12 2 10 4 12
4 12 3 11 4 12 3 11 6 14 5 13 6 14 5 13 1 9 3 11 1 9 3 11
6 14 5 13 6 14 5 13 4 12 3 11 4 12 3 11 6 14 8 16 6 14 8 16
8 16 7 15 8 16 7 15 8 16 7 15 8 16 7 15 5 13 7 15 5 13 7 15
      46/S 46/C          47/S 47/C          48/S 48/C
2 10 4 12 2 10 4 12 2 10 6 14 2 10 6 14 2 10 6 14 2 10 6 14
6 14 8 16 6 14 8 16 1 9 5 13 1 9 5 13 4 12 8 16 4 12 8 16
1 9 3 11 1 9 3 11 4 12 8 16 4 12 8 16 1 9 5 13 1 9 5 13
5 13 7 15 5 13 7 15 3 11 7 15 3 11 7 15 3 11 7 15 3 11 7 15
      49/S 49/C          50/S 50/C          51/S 51/C
3 1 4 2 3 1 4 2 3 1 4 2 3 1 4 2 3 1 7 5 3 1 7 5
7 5 8 6 7 5 8 6 11 9 12 10 11 9 12 10 4 2 8 6 4 2 8 6
11 9 12 10 11 9 12 10 7 5 8 6 7 5 8 6 11 9 15 13 11 9 15 13
15 13 16 14 15 13 16 14 15 13 16 14 15 13 16 14 12 10 16 14 12 10 16 14

```

[Counts = 384/S : 384/C]

ひな形を変え作り方を変えても、出来る原初四方陣の解集合は全く同じで、解の出る順番がいささか変わるだけ。だから、原初四方陣は、共通の解集合を1つ用意すればよい。

その各個について下のような「ひな形変換法」を施し、384個の目的方陣を作る。

```

/**/
/* ひな形変換法 */
/* 原初四方陣/ 対称四方陣/ 完全四方陣/
|-----|-----|-----|
| 1| 2| 3| 4| | 1|15|14| 4| | 1|15| 4|14|
|-----|-----|-----|
| 5| 6| 7| 8| | 12| 6| 7| 9| | 12| 6| 9| 7|
|-----|-----|-----|
| 9|10|11|12| | 8|10|11| 5| | 13| 3|16| 2|
|-----|-----|-----|
|13|14|15|16| | 13| 3| 2|16| | 8|10| 5|11|
|-----|-----|-----|
*/
/* 対称四方陣用のひな形変換法 */
void damts(void){
    dm[1]=nm[1];   dm[2]=nm[15];   dm[3]=nm[14];   dm[4]=nm[4];
    dm[5]=nm[12];  dm[6]=nm[6];   dm[7]=nm[7];   dm[8]=nm[9];
    dm[9]=nm[8];   dm[10]=nm[10];  dm[11]=nm[11]; dm[12]=nm[5];
    dm[13]=nm[13]; dm[14]=nm[3];   dm[15]=nm[2];  dm[16]=nm[16];
}
/**/
/* 完全四方陣用のひな形変換法 */
void damtc(void){
    dm[1]=nm[1];   dm[2]=nm[15];   dm[3]=nm[4];   dm[4]=nm[14];
    dm[5]=nm[12];  dm[6]=nm[6];   dm[7]=nm[9];   dm[8]=nm[7];
    dm[9]=nm[13];  dm[10]=nm[3];  dm[11]=nm[16]; dm[12]=nm[2];
    dm[13]=nm[8];  dm[14]=nm[10]; dm[15]=nm[5];  dm[16]=nm[11];
}
/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Primitive Set] */
void prp2trios(short x){
    short l,l4,m,n,p;

```

```

for(m=0;m<x;m=m+2){
  for(n=1;n<17;n++){nm[n]=pts[m][n];}
  damts(); for(n=1;n<17;n++){d[0][n]=dm[n];}
  for(n=1;n<17;n++){nm[n]=ptc[m][n];}
  damtc(); for(n=1;n<17;n++){d[1][n]=dm[n];}
  for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
  damts(); for(n=1;n<17;n++){d[2][n]=dm[n];}
  for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
  damtc(); for(n=1;n<17;n++){d[3][n]=dm[n];}
  printf("%10d/P      /S      /C%12d/P      /S      /C\n",
    pts[m][0],pts[m+1][0]);
  for(l=0;l<4;l++){l4=l*4;
    for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
    printf(" ");
    for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
    printf(" ");
    for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
    printf(" ");
    for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
    printf(" ");
    for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
    printf(" ");
    for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
    printf("\n");
  }
}
}
}
/**/

```

**** 共通の原初四方陣から2つのタイプのひな形変換法によって目的方陣を構成する実験-1 ****

*** 384個の原始解レベルで：原初四方陣/P, 補数対称四方陣/S, 完全型汎四方陣/C ***

1/P				/S				/C				2/P				/S				/C			
1	2	3	4	1	15	14	4	1	15	4	14	1	2	3	4	1	15	14	4	1	15	4	14
5	6	7	8	12	6	7	9	12	6	9	7	9	10	11	12	8	10	11	5	8	10	5	11
9	10	11	12	8	10	11	5	13	3	16	2	5	6	7	8	12	6	7	9	13	3	16	2
13	14	15	16	13	3	2	16	8	10	5	11	13	14	15	16	13	3	2	16	12	6	9	7
3/P				/S				/C				4/P				/S				/C			
1	2	5	6	1	15	12	6	1	15	6	12	1	2	5	6	1	15	12	6	1	15	6	12
3	4	7	8	14	4	7	9	14	4	9	7	9	10	13	14	8	10	13	3	8	10	3	13
9	10	13	14	8	10	13	3	11	5	16	2	3	4	7	8	14	4	7	9	11	5	16	2
11	12	15	16	11	5	2	16	8	10	3	13	11	12	15	16	11	5	2	16	14	4	9	7
5/P				/S				/C				6/P				/S				/C			
1	2	9	10	1	15	8	10	1	15	10	8	1	2	9	10	1	15	8	10	1	15	10	8
3	4	11	12	14	4	11	5	14	4	5	11	5	6	13	14	12	6	13	3	12	6	3	13
5	6	13	14	12	6	13	3	7	9	16	2	3	4	11	12	14	4	11	5	7	9	16	2
7	8	15	16	7	9	2	16	12	6	3	13	7	8	15	16	7	9	2	16	14	4	5	11
7/P				/S				/C				8/P				/S				/C			
1	3	2	4	1	14	15	4	1	14	4	15	1	3	2	4	1	14	15	4	1	14	4	15
5	7	6	8	12	7	6	9	12	7	9	6	9	11	10	12	8	11	10	5	8	11	5	10
9	11	10	12	8	11	10	5	13	2	16	3	5	7	6	8	12	7	6	9	13	2	16	3
13	15	14	16	13	2	3	16	8	11	5	10	13	15	14	16	13	2	3	16	12	7	9	6
9/P				/S				/C				10/P				/S				/C			
1	3	5	7	1	14	12	7	1	14	7	12	1	3	5	7	1	14	12	7	1	14	7	12
2	4	6	8	15	4	6	9	15	4	9	6	9	11	13	15	8	11	13	2	8	11	2	13
9	11	13	15	8	11	13	2	10	5	16	3	2	4	6	8	15	4	6	9	10	5	16	3
10	12	14	16	10	5	3	16	8	11	2	13	10	12	14	16	10	5	3	16	15	4	9	6
11/P				/S				/C				12/P				/S				/C			

1 3 9 11	1 14 8 11	1 14 11 8	1 3 9 11	1 14 8 11	1 14 11 8
2 4 10 12	15 4 10 5	15 4 5 10	5 7 13 15	12 7 13 2	12 7 2 13
5 7 13 15	12 7 13 2	6 9 16 3	2 4 10 12	15 4 10 5	6 9 16 3
6 8 14 16	6 9 3 16	12 7 2 13	6 8 14 16	6 9 3 16	15 4 5 10
13/P	/S	/C	14/P	/S	/C
1 5 2 6	1 12 15 6	1 12 6 15	1 5 2 6	1 12 15 6	1 12 6 15
3 7 4 8	14 7 4 9	14 7 9 4	9 13 10 14	8 13 10 3	8 13 3 10
9 13 10 14	8 13 10 3	11 2 16 5	3 7 4 8	14 7 4 9	11 2 16 5
11 15 12 16	11 2 5 16	8 13 3 10	11 15 12 16	11 2 5 16	14 7 9 4
15/P	/S	/C	16/P	/S	/C
1 5 3 7	1 12 14 7	1 12 7 14	1 5 3 7	1 12 14 7	1 12 7 14
2 6 4 8	15 6 4 9	15 6 9 4	9 13 11 15	8 13 11 2	8 13 2 11
9 13 11 15	8 13 11 2	10 3 16 5	2 6 4 8	15 6 4 9	10 3 16 5
10 14 12 16	10 3 5 16	8 13 2 11	10 14 12 16	10 3 5 16	15 6 9 4
17/P	/S	/C	18/P	/S	/C
1 5 9 13	1 12 8 13	1 12 13 8	1 5 9 13	1 12 8 13	1 12 13 8
2 6 10 14	15 6 10 3	15 6 3 10	3 7 11 15	14 7 11 2	14 7 2 11
3 7 11 15	14 7 11 2	4 9 16 5	2 6 10 14	15 6 10 3	4 9 16 5
4 8 12 16	4 9 5 16	14 7 2 11	4 8 12 16	4 9 5 16	15 6 3 10
19/P	/S	/C	20/P	/S	/C
1 9 2 10	1 8 15 10	1 8 10 15	1 9 2 10	1 8 15 10	1 8 10 15
3 11 4 12	14 11 4 5	14 11 5 4	5 13 6 14	12 13 6 3	12 13 3 6
5 13 6 14	12 13 6 3	7 2 16 9	3 11 4 12	14 11 4 5	7 2 16 9
7 15 8 16	7 2 9 16	12 13 3 6	7 15 8 16	7 2 9 16	14 11 5 4
21/P	/S	/C	22/P	/S	/C
1 9 3 11	1 8 14 11	1 8 11 14	1 9 3 11	1 8 14 11	1 8 11 14
2 10 4 12	15 10 4 5	15 10 5 4	5 13 7 15	12 13 7 2	12 13 2 7
5 13 7 15	12 13 7 2	6 3 16 9	2 10 4 12	15 10 4 5	6 3 16 9
6 14 8 16	6 3 9 16	12 13 2 7	6 14 8 16	6 3 9 16	15 10 5 4
23/P	/S	/C	24/P	/S	/C
1 9 5 13	1 8 12 13	1 8 13 12	1 9 5 13	1 8 12 13	1 8 13 12
2 10 6 14	15 10 6 3	15 10 3 6	3 11 7 15	14 11 7 2	14 11 2 7
3 11 7 15	14 11 7 2	4 5 16 9	2 10 6 14	15 10 6 3	4 5 16 9
4 12 8 16	4 5 9 16	14 11 2 7	4 12 8 16	4 5 9 16	15 10 3 6
25/P	/S	/C	26/P	/S	/C
2 1 4 3	2 16 13 3	2 16 3 13	2 1 4 3	2 16 13 3	2 16 3 13
6 5 8 7	11 5 8 10	11 5 10 8	10 9 12 11	7 9 12 6	7 9 6 12
10 9 12 11	7 9 12 6	14 4 15 1	6 5 8 7	11 5 8 10	14 4 15 1
14 13 16 15	14 4 1 15	7 9 6 12	14 13 16 15	14 4 1 15	11 5 10 8
27/P	/S	/C	28/P	/S	/C
2 1 6 5	2 16 11 5	2 16 5 11	2 1 6 5	2 16 11 5	2 16 5 11
4 3 8 7	13 3 8 10	13 3 10 8	10 9 14 13	7 9 14 4	7 9 4 14
10 9 14 13	7 9 14 4	12 6 15 1	4 3 8 7	13 3 8 10	12 6 15 1
12 11 16 15	12 6 1 15	7 9 4 14	12 11 16 15	12 6 1 15	13 3 10 8
29/P	/S	/C	30/P	/S	/C
2 1 10 9	2 16 7 9	2 16 9 7	2 1 10 9	2 16 7 9	2 16 9 7
4 3 12 11	13 3 12 6	13 3 6 12	6 5 14 13	11 5 14 4	11 5 4 14
6 5 14 13	11 5 14 4	8 10 15 1	4 3 12 11	13 3 12 6	8 10 15 1
8 7 16 15	8 10 1 15	11 5 4 14	8 7 16 15	8 10 1 15	13 3 6 12
31/P	/S	/C	32/P	/S	/C
2 4 1 3	2 13 16 3	2 13 3 16	2 4 1 3	2 13 16 3	2 13 3 16

6 8 5 7	11 8 5 10	11 8 10 5	10 12 9 11	7 12 9 6	7 12 6 9
10 12 9 11	7 12 9 6	14 1 15 4	6 8 5 7	11 8 5 10	14 1 15 4
14 16 13 15	14 1 4 15	7 12 6 9	14 16 13 15	14 1 4 15	11 8 10 5
33/P	/S	/C	34/P	/S	/C
2 4 6 8	2 13 11 8	2 13 8 11	2 4 6 8	2 13 11 8	2 13 8 11
1 3 5 7	16 3 5 10	16 3 10 5	10 12 14 16	7 12 14 1	7 12 1 14
10 12 14 16	7 12 14 1	9 6 15 4	1 3 5 7	16 3 5 10	9 6 15 4
9 11 13 15	9 6 4 15	7 12 1 14	9 11 13 15	9 6 4 15	16 3 10 5
35/P	/S	/C	36/P	/S	/C
2 4 10 12	2 13 7 12	2 13 12 7	2 4 10 12	2 13 7 12	2 13 12 7
1 3 9 11	16 3 9 6	16 3 6 9	6 8 14 16	11 8 14 1	11 8 1 14
6 8 14 16	11 8 14 1	5 10 15 4	1 3 9 11	16 3 9 6	5 10 15 4
5 7 13 15	5 10 4 15	11 8 1 14	5 7 13 15	5 10 4 15	16 3 6 9
37/P	/S	/C	38/P	/S	/C
2 6 1 5	2 11 16 5	2 11 5 16	2 6 1 5	2 11 16 5	2 11 5 16
4 8 3 7	13 8 3 10	13 8 10 3	10 14 9 13	7 14 9 4	7 14 4 9
10 14 9 13	7 14 9 4	12 1 15 6	4 8 3 7	13 8 3 10	12 1 15 6
12 16 11 15	12 1 6 15	7 14 4 9	12 16 11 15	12 1 6 15	13 8 10 3
39/P	/S	/C	40/P	/S	/C
2 6 4 8	2 11 13 8	2 11 8 13	2 6 4 8	2 11 13 8	2 11 8 13
1 5 3 7	16 5 3 10	16 5 10 3	10 14 12 16	7 14 12 1	7 14 1 12
10 14 12 16	7 14 12 1	9 4 15 6	1 5 3 7	16 5 3 10	9 4 15 6
9 13 11 15	9 4 6 15	7 14 1 12	9 13 11 15	9 4 6 15	16 5 10 3
41/P	/S	/C	42/P	/S	/C
2 6 10 14	2 11 7 14	2 11 14 7	2 6 10 14	2 11 7 14	2 11 14 7
1 5 9 13	16 5 9 4	16 5 4 9	4 8 12 16	13 8 12 1	13 8 1 12
4 8 12 16	13 8 12 1	3 10 15 6	1 5 9 13	16 5 9 4	3 10 15 6
3 7 11 15	3 10 6 15	13 8 1 12	3 7 11 15	3 10 6 15	16 5 4 9
43/P	/S	/C	44/P	/S	/C
2 10 1 9	2 7 16 9	2 7 9 16	2 10 1 9	2 7 16 9	2 7 9 16
4 12 3 11	13 12 3 6	13 12 6 3	6 14 5 13	11 14 5 4	11 14 4 5
6 14 5 13	11 14 5 4	8 1 15 10	4 12 3 11	13 12 3 6	8 1 15 10
8 16 7 15	8 1 10 15	11 14 4 5	8 16 7 15	8 1 10 15	13 12 6 3
45/P	/S	/C	46/P	/S	/C
2 10 4 12	2 7 13 12	2 7 12 13	2 10 4 12	2 7 13 12	2 7 12 13
1 9 3 11	16 9 3 6	16 9 6 3	6 14 8 16	11 14 8 1	11 14 1 8
6 14 8 16	11 14 8 1	5 4 15 10	1 9 3 11	16 9 3 6	5 4 15 10
5 13 7 15	5 4 10 15	11 14 1 8	5 13 7 15	5 4 10 15	16 9 6 3
47/P	/S	/C	48/P	/S	/C
2 10 6 14	2 7 11 14	2 7 14 11	2 10 6 14	2 7 11 14	2 7 14 11
1 9 5 13	16 9 5 4	16 9 4 5	4 12 8 16	13 12 8 1	13 12 1 8
4 12 8 16	13 12 8 1	3 6 15 10	1 9 5 13	16 9 5 4	3 6 15 10
3 11 7 15	3 6 10 15	13 12 1 8	3 11 7 15	3 6 10 15	16 9 4 5
49/P	/S	/C	50/P	/S	/C
3 1 4 2	3 16 13 2	3 16 2 13	3 1 4 2	3 16 13 2	3 16 2 13
7 5 8 6	10 5 8 11	10 5 11 8	11 9 12 10	6 9 12 7	6 9 7 12
11 9 12 10	6 9 12 7	15 4 14 1	7 5 8 6	10 5 8 11	15 4 14 1
15 13 16 14	15 4 1 14	6 9 7 12	15 13 16 14	15 4 1 14	10 5 11 8

[Counts = 384/P, 384/S, 384/C]

2-2. 標準解の出力

48 個の「標準解」のみを出力させたい時には、次のような不等式を用いる。反転・回転形を整理し、求める解集合をふるい出す。

リスト出力用の不等式： $n1 < n16$; $n1 < n4$; $n1 < n13$; $n2 > n5$;

もし原初方阵を作る最初の段階からして 48 個に絞りたいのであれば、そのための不等式は、「ひな形」に敬意を表して、次のようにしなければならない。

標準解のリスト出力用の不等式： $n1 < n11$; $n1 < n8$; $n1 < n14$; $n15 > n12$;

```

/**/
/* Make 'Prototype Squares' 4x4 and Apply 'DAMT' */
/*
    [基本図]          [ひな形]
15 16 13 14 15 16 13 14   5 11 14  4  5 11 14  4
3  4 | 1 | 2 | 3 | 4 | 1  2   10 8 | 1 |15|10| 8 | 1 15
7  8 | 5 | 6 | 7 | 8 | 5  6   3 13|12| 6 | 3 |13|12  6
11 12| 9 |10|11|12| 9 10   16  2| 7 | 9 |16| 2 | 7  9
15 16|13|14|15|16|13 14   5 11|14| 4 | 5 |11|14  4
3  4  1  2  3  4  1  2   10 8  1 15 10  8  1 15
    [基本条件式]
n1+n15+n10+n8=34 ... rw1; | n1+n12+n7+n14 ... c11;
n12+n6+n3+n13=34 ... rw2; | n15+n6+n9+n4 ... c12;
n7+n9+n16+n2=34 ... rw3; | n10+n3+n16+n5 ... c13;
n14+n4+n5+n11=34 ... rw4; | n8+n13+n2+n11 ... c14;
n1+n16=n2+n15=n3+n14=n4+n13=n5+n12=n6=n11=n7+n10=n8+n9=17 ... cmlt
n1<n14; n1<n8; n1<n11; and n15<n12;
*/
/* ひな形変換法：Prototype into 'Complete' Object */
void damt(void){
    dnm[1]=nm[1];   dnm[2]=nm[15];   dnm[3]=nm[10];   dnm[4]=nm[8];
    dnm[5]=nm[12];   dnm[6]=nm[6];    dnm[7]=nm[3];    dnm[8]=nm[13];
    dnm[9]=nm[7];    dnm[10]=nm[9];   dnm[11]=nm[16];  dnm[12]=nm[2];
    dnm[13]=nm[14]; dnm[14]=nm[4];   dnm[15]=nm[5];   dnm[16]=nm[11];
}
/**/

```

試みに新たなひな形と変換法を用意し、それに準拠した原初方阵を構成してみよう。

次のリストが、その原初四方陣とひな形変換した完全型汎四方陣の標準解 48 個である。

対応する原初四方陣とカップリングして表示してみた。

最後に完全型四方陣の古いリストと比較対照したモニター結果も添えてみた。

* 新しい「ひな形変換法」と原初四方陣によって再構成した完全汎四方陣の標準解 48 個 *

1/P	6/C	2/P	4/C	3/P	5/C
1 2 3 4	1 15 10 8	1 2 3 4	1 15 6 12	1 2 5 6	1 15 10 8
5 6 7 8	12 6 3 13	9 10 11 12	8 10 3 13	3 4 7 8	14 4 5 11
9 10 11 12	7 9 16 2	5 6 7 8	11 5 16 2	9 10 13 14	7 9 16 2
13 14 15 16	14 4 5 11	13 14 15 16	14 4 9 7	11 12 15 16	12 6 3 13
4/P	2/C	5/P	3/C	6/P	1/C
1 2 5 6	1 15 4 14	1 2 9 10	1 15 6 12	1 2 9 10	1 15 4 14
9 10 13 14	8 10 5 11	3 4 11 12	14 4 9 7	5 6 13 14	12 6 9 7
3 4 7 8	13 3 16 2	5 6 13 14	11 5 16 2	3 4 11 12	13 3 16 2
11 12 15 16	12 6 9 7	7 8 15 16	8 10 3 13	7 8 15 16	8 10 5 11

	7/P	10/C		8/P	9/C		9/P	8/C						
1	3	2 4	1 14	11 8	1	3	2 4	1 14	7 12	1	3	5 7	1 14	4 15
5	7	6 8	12 7	2 13	9	11	10 12	8 11	2 13	9	11	13 15	8 11	5 10
9	11	10 12	6 9	16 3	5	7	6 8	10 5	16 3	2	4	6 8	13 2	16 3
13	15	14 16	15 4	5 10	13	15	14 16	15 4	9 6	10	12	14 16	12 7	9 6
	10/P	7/C		11/P	12/C		12/P	11/C						
1	3	9 11	1 14	4 15	1	5	2 6	1 12	7 14	1	5	3 7	1 12	6 15
5	7	13 15	12 7	9 6	9	13	10 14	8 13	2 11	9	13	11 15	8 13	3 10
2	4	10 12	13 2	16 3	3	7	4 8	10 3	16 5	2	6	4 8	11 2	16 5
6	8	14 16	8 11	5 10	11	15	12 16	15 6	9 4	10	14	12 16	14 7	9 4
	13/P	18/C		14/P	16/C		15/P	17/C						
2	1	4 3	2 16	9 7	2	1	4 3	2 16	5 11	2	1	6 5	2 16	9 7
6	5	8 7	11 5	4 14	10	9	12 11	7 9	4 14	4	3	8 7	13 3	6 12
10	9	12 11	8 10	15 1	6	5	8 7	12 6	15 1	10	9	14 13	8 10	15 1
14	13	16 15	13 3	6 12	14	13	16 15	13 3	10 8	12	11	16 15	11 5	4 14
	16/P	14/C		17/P	15/C		18/P	13/C						
2	1	6 5	2 16	3 13	2	1	10 9	2 16	5 11	2	1	10 9	2 16	3 13
10	9	14 13	7 9	6 12	4	3	12 11	13 3	10 8	6	5	14 13	11 5	10 8
4	3	8 7	14 4	15 1	6	5	14 13	12 6	15 1	4	3	12 11	14 4	15 1
12	11	16 15	11 5	10 8	8	7	16 15	7 9	4 14	8	7	16 15	7 9	6 12
	19/P	22/C		20/P	21/C		21/P	20/C						
2	4	1 3	2 13	12 7	2	4	1 3	2 13	8 11	2	4	6 8	2 13	3 16
6	8	5 7	11 8	1 14	10	12	9 11	7 12	1 14	10	12	14 16	7 12	6 9
10	12	9 11	5 10	15 4	6	8	5 7	9 6	15 4	1	3	5 7	14 1	15 4
14	16	13 15	16 3	6 9	14	16	13 15	16 3	10 5	9	11	13 15	11 8	10 5
	22/P	19/C		23/P	24/C		24/P	23/C						
2	4	10 12	2 13	3 16	2	6	1 5	2 11	8 13	2	6	4 8	2 11	5 16
6	8	14 16	11 8	10 5	10	14	9 13	7 14	1 12	10	14	12 16	7 14	4 9
1	3	9 11	14 1	15 4	4	8	3 7	9 4	15 6	1	5	3 7	12 1	15 6
5	7	13 15	7 12	6 9	12	16	11 15	16 5	10 3	9	13	11 15	13 8	10 3
	25/P	30/C		26/P	28/C		27/P	29/C						
3	1	4 2	3 16	9 6	3	1	4 2	3 16	5 10	3	1	7 5	3 16	9 6
7	5	8 6	10 5	4 15	11	9	12 10	6 9	4 15	4	2	8 6	13 2	7 12
11	9	12 10	8 11	14 1	7	5	8 6	12 7	14 1	11	9	15 13	8 11	14 1
15	13	16 14	13 2	7 12	15	13	16 14	13 2	11 8	12	10	16 14	10 5	4 15
	28/P	26/C		29/P	27/C		30/P	25/C						
3	1	7 5	3 16	2 13	3	1	11 9	3 16	5 10	3	1	11 9	3 16	2 13
11	9	15 13	6 9	7 12	4	2	12 10	13 2	11 8	7	5	15 13	10 5	11 8
4	2	8 6	15 4	14 1	7	5	15 13	12 7	14 1	4	2	12 10	15 4	14 1
12	10	16 14	10 5	11 8	8	6	16 14	6 9	4 15	8	6	16 14	6 9	7 12
	31/P	34/C		32/P	33/C		33/P	32/C						
3	4	1 2	3 13	12 6	3	4	1 2	3 13	8 10	3	4	7 8	3 13	2 16
7	8	5 6	10 8	1 15	11	12	9 10	6 12	1 15	11	12	15 16	6 12	7 9
11	12	9 10	5 11	14 4	7	8	5 6	9 7	14 4	1	2	5 6	15 1	14 4
15	16	13 14	16 2	7 9	15	16	13 14	16 2	11 5	9	10	13 14	10 8	11 5
	34/P	31/C		35/P	40/C		36/P	38/C						
3	4	11 12	3 13	2 16	4	2	3 1	4 15	10 5	4	2	3 1	4 15	6 9
7	8	15 16	10 8	11 5	8	6	7 5	9 6	3 16	12	10	11 9	5 10	3 16
1	2	9 10	15 1	14 4	12	10	11 9	7 12	13 2	8	6	7 5	11 8	13 2
5	6	13 14	6 12	7 9	16	14	15 13	14 1	8 11	16	14	15 13	14 1	12 7
	37/P	39/C		38/P	36/C		39/P	37/C						
4	2	8 6	4 15	10 5	4	2	8 6	4 15	1 14	4	2	12 10	4 15	6 9

めに用いた不等式に抵触してしまい、必ずしも相互に一対一対応しなくなるからだ。
あくまで384個の原始解の間で、型変換を行う必要がある。

次のリストは、原初四方陣384個をひな形変換して先ず対称四方陣384個を作り、それを型変換して完全四方陣384個を作った実験の報告である（リストは抜粋のみ）。

*** 原初四方陣とひな形変換法・型変換によって対称型と完全型の四方陣を構成する実験 ***

[1] 1/P 1/S 1/C	[2] 2/P 2/S 2/C
1 2 3 4 1 15 14 4 1 15 4 14	1 2 3 4 1 15 14 4 1 15 4 14
5 6 7 8 12 6 7 9 12 6 9 7	9 10 11 12 8 10 11 5 8 10 5 11
9 10 11 12 8 10 11 5 13 3 16 2	5 6 7 8 12 6 7 9 13 3 16 2
13 14 15 16 13 3 2 16 8 10 5 11	13 14 15 16 13 3 2 16 12 6 9 7
[3] 3/P 3/S 3/C	[4] 4/P 4/S 4/C
1 2 5 6 1 15 12 6 1 15 6 12	1 2 5 6 1 15 12 6 1 15 6 12
3 4 7 8 14 4 7 9 14 4 9 7	9 10 13 14 8 10 13 3 8 10 3 13
9 10 13 14 8 10 13 3 11 5 16 2	3 4 7 8 14 4 7 9 11 5 16 2
11 12 15 16 11 5 2 16 8 10 3 13	11 12 15 16 11 5 2 16 14 4 9 7
[5] 5/P 5/S 5/C	[6] 6/P 6/S 6/C
1 2 9 10 1 15 8 10 1 15 10 8	1 2 9 10 1 15 8 10 1 15 10 8
3 4 11 12 14 4 11 5 14 4 5 11	5 6 13 14 12 6 13 3 12 6 3 13
5 6 13 14 12 6 13 3 7 9 16 2	3 4 11 12 14 4 11 5 7 9 16 2
7 8 15 16 7 9 2 16 12 6 3 13	7 8 15 16 7 9 2 16 14 4 5 11
[7] 7/P 7/S 7/C	[8] 8/P 8/S 8/C
1 3 2 4 1 14 15 4 1 14 4 15	1 3 2 4 1 14 15 4 1 14 4 15
5 7 6 8 12 7 6 9 12 7 9 6	9 11 10 12 8 11 10 5 8 11 5 10
9 11 10 12 8 11 10 5 13 2 16 3	5 7 6 8 12 7 6 9 13 2 16 3
13 15 14 16 13 2 3 16 8 11 5 10	13 15 14 16 13 2 3 16 12 7 9 6
[9] 9/P 9/S 9/C	[10] 10/P 10/S 10/C
1 3 5 7 1 14 12 7 1 14 7 12	1 3 5 7 1 14 12 7 1 14 7 12
2 4 6 8 15 4 6 9 15 4 9 6	9 11 13 15 8 11 13 2 8 11 2 13
9 11 13 15 8 11 13 2 10 5 16 3	2 4 6 8 15 4 6 9 10 5 16 3
10 12 14 16 10 5 3 16 8 11 2 13	10 12 14 16 10 5 3 16 15 4 9 6
[11] 11/P 11/S 11/C	[12] 12/P 12/S 12/C
1 3 9 11 1 14 8 11 1 14 11 8	1 3 9 11 1 14 8 11 1 14 11 8
2 4 10 12 15 4 10 5 15 4 5 10	5 7 13 15 12 7 13 2 12 7 2 13
5 7 13 15 12 7 13 2 6 9 16 3	2 4 10 12 15 4 10 5 6 9 16 3
6 8 14 16 6 9 3 16 12 7 2 13	6 8 14 16 6 9 3 16 15 4 5 10
[13] 13/P 13/S 13/C	[14] 14/P 14/S 14/C
1 5 2 6 1 12 15 6 1 12 6 15	1 5 2 6 1 12 15 6 1 12 6 15
3 7 4 8 14 7 4 9 14 7 9 4	9 13 10 14 8 13 10 3 8 13 3 10
9 13 10 14 8 13 10 3 11 2 16 5	3 7 4 8 14 7 4 9 11 2 16 5
11 15 12 16 11 2 5 16 8 13 3 10	11 15 12 16 11 2 5 16 14 7 9 4
[15] 15/P 15/S 15/C	[16] 16/P 16/S 16/C
1 5 3 7 1 12 14 7 1 12 7 14	1 5 3 7 1 12 14 7 1 12 7 14
2 6 4 8 15 6 4 9 15 6 9 4	9 13 11 15 8 13 11 2 8 13 2 11
9 13 11 15 8 13 11 2 10 3 16 5	2 6 4 8 15 6 4 9 10 3 16 5
10 14 12 16 10 3 5 16 8 13 2 11	10 14 12 16 10 3 5 16 15 6 9 4
[17] 17/P 17/S 17/C	[18] 18/P 18/S 18/C
1 5 9 13 1 12 8 13 1 12 13 8	1 5 9 13 1 12 8 13 1 12 13 8
2 6 10 14 15 6 10 3 15 6 3 10	3 7 11 15 14 7 11 2 14 7 2 11
3 7 11 15 14 7 11 2 4 9 16 5	2 6 10 14 15 6 10 3 4 9 16 5
4 8 12 16 4 9 5 16 14 7 2 11	4 8 12 16 4 9 5 16 15 6 3 10

	5/P	5/S	5/C	6/P	6/S	6/C
1	2 9 10	1 15 8 10	1 15 10 8	1 2 9 10	1 15 8 10	1 15 10 8
3	4 11 12	14 4 11 5	14 4 5 11	5 6 13 14	12 6 13 3	12 6 3 13
5	6 13 14	12 6 13 3	7 9 16 2	3 4 11 12	14 4 11 5	7 9 16 2
7	8 15 16	7 9 2 16	12 6 3 13	7 8 15 16	7 9 2 16	14 4 5 11
	7/P	7/S	7/C	8/P	8/S	8/C
1	3 2 4	1 14 15 4	1 14 4 15	1 3 2 4	1 14 15 4	1 14 4 15
5	7 6 8	12 7 6 9	12 7 9 6	9 11 10 12	8 11 10 5	8 11 5 10
9	11 10 12	8 11 10 5	13 2 16 3	5 7 6 8	12 7 6 9	13 2 16 3
13	15 14 16	13 2 3 16	8 11 5 10	13 15 14 16	13 2 3 16	12 7 9 6
	9/P	0/S	0/C	10/P	9/S	9/C
1	3 5 7	0 0 0 0	0 0 0 0	1 3 5 7	1 14 12 7	1 14 7 12
2	4 6 8	0 0 0 0	0 0 0 0	9 11 13 15	8 11 13 2	8 11 2 13
9	11 13 15	0 0 0 0	0 0 0 0	2 4 6 8	15 4 6 9	10 5 16 3
10	12 14 16	0 0 0 0	0 0 0 0	10 12 14 16	10 5 3 16	15 4 9 6
	11/P	0/S	0/C	12/P	10/S	10/C
1	3 9 11	0 0 0 0	0 0 0 0	1 3 9 11	1 14 8 11	1 14 11 8
2	4 10 12	0 0 0 0	0 0 0 0	5 7 13 15	12 7 13 2	12 7 2 13
5	7 13 15	0 0 0 0	0 0 0 0	2 4 10 12	15 4 10 5	6 9 16 3
6	8 14 16	0 0 0 0	0 0 0 0	6 8 14 16	6 9 3 16	15 4 5 10
	13/P	0/S	0/C	14/P	11/S	11/C
1	5 2 6	0 0 0 0	0 0 0 0	1 5 2 6	1 12 15 6	1 12 6 15
3	7 4 8	0 0 0 0	0 0 0 0	9 13 10 14	8 13 10 3	8 13 3 10
9	13 10 14	0 0 0 0	0 0 0 0	3 7 4 8	14 7 4 9	11 2 16 5
11	15 12 16	0 0 0 0	0 0 0 0	11 15 12 16	11 2 5 16	14 7 9 4
	15/P	0/S	0/C	16/P	12/S	12/C
1	5 3 7	0 0 0 0	0 0 0 0	1 5 3 7	1 12 14 7	1 12 7 14
2	6 4 8	0 0 0 0	0 0 0 0	9 13 11 15	8 13 11 2	8 13 2 11
9	13 11 15	0 0 0 0	0 0 0 0	2 6 4 8	15 6 4 9	10 3 16 5
10	14 12 16	0 0 0 0	0 0 0 0	10 14 12 16	10 3 5 16	15 6 9 4
	17/P	0/S	0/C	18/P	0/S	0/C
	19/P	0/S	0/C	20/P	0/S	0/C
	21/P	0/S	0/C	22/P	0/S	0/C
	23/P	0/S	0/C	24/P	0/S	0/C
	25/P	13/S	13/C	26/P	14/S	14/C
2	1 4 3	2 16 13 3	2 16 3 13	2 1 4 3	2 16 13 3	2 16 3 13
6	5 8 7	11 5 8 10	11 5 10 8	10 9 12 11	7 9 12 6	7 9 6 12
10	9 12 11	7 9 12 6	14 4 15 1	6 5 8 7	11 5 8 10	14 4 15 1
14	13 16 15	14 4 1 15	7 9 6 12	14 13 16 15	14 4 1 15	11 5 10 8
	27/P	15/S	15/C	28/P	16/S	16/C
2	1 6 5	2 16 11 5	2 16 5 11	2 1 6 5	2 16 11 5	2 16 5 11
4	3 8 7	13 3 8 10	13 3 10 8	10 9 14 13	7 9 14 4	7 9 4 14
10	9 14 13	7 9 14 4	12 6 15 1	4 3 8 7	13 3 8 10	12 6 15 1
12	11 16 15	12 6 1 15	7 9 4 14	12 11 16 15	12 6 1 15	13 3 10 8
	29/P	17/S	17/C	30/P	18/S	18/C
2	1 10 9	2 16 7 9	2 16 9 7	2 1 10 9	2 16 7 9	2 16 9 7
4	3 12 11	13 3 12 6	13 3 6 12	6 5 14 13	11 5 14 4	11 5 4 14
6	5 14 13	11 5 14 4	8 10 15 1	4 3 12 11	13 3 12 6	8 10 15 1
8	7 16 15	8 10 1 15	11 5 4 14	8 7 16 15	8 10 1 15	13 3 6 12
	31/P	19/S	19/C	32/P	20/S	20/C
2	4 1 3	2 13 16 3	2 13 3 16	2 4 1 3	2 13 16 3	2 13 3 16
6	8 5 7	11 8 5 10	11 8 10 5	10 12 9 11	7 12 9 6	7 12 6 9

10 12 9 11	7 12 9 6	14 1 15 4	6 8 5 7	11 8 5 10	14 1 15 4
14 16 13 15	14 1 4 15	7 12 6 9	14 16 13 15	14 1 4 15	11 8 10 5
33/P	0/S	0/C	34/P	21/S	21/C
2 4 6 8	0 0 0 0	0 0 0 0	2 4 6 8	2 13 11 8	2 13 8 11
1 3 5 7	0 0 0 0	0 0 0 0	10 12 14 16	7 12 14 1	7 12 1 14
10 12 14 16	0 0 0 0	0 0 0 0	1 3 5 7	16 3 5 10	9 6 15 4
9 11 13 15	0 0 0 0	0 0 0 0	9 11 13 15	9 6 4 15	16 3 10 5
35/P	0/S	0/C	36/P	22/S	22/C
2 4 10 12	0 0 0 0	0 0 0 0	2 4 10 12	2 13 7 12	2 13 12 7
1 3 9 11	0 0 0 0	0 0 0 0	6 8 14 16	11 8 14 1	11 8 1 14
6 8 14 16	0 0 0 0	0 0 0 0	1 3 9 11	16 3 9 6	5 10 15 4
5 7 13 15	0 0 0 0	0 0 0 0	5 7 13 15	5 10 4 15	16 3 6 9
37/P	0/S	0/C	38/P	23/S	23/C
2 6 1 5	0 0 0 0	0 0 0 0	2 6 1 5	2 11 16 5	2 11 5 16
4 8 3 7	0 0 0 0	0 0 0 0	10 14 9 13	7 14 9 4	7 14 4 9
10 14 9 13	0 0 0 0	0 0 0 0	4 8 3 7	13 8 3 10	12 1 15 6
12 16 11 15	0 0 0 0	0 0 0 0	12 16 11 15	12 1 6 15	13 8 10 3
39/P	0/S	0/C	40/P	24/S	24/C
2 6 4 8	0 0 0 0	0 0 0 0	2 6 4 8	2 11 13 8	2 11 8 13
1 5 3 7	0 0 0 0	0 0 0 0	10 14 12 16	7 14 12 1	7 14 1 12
10 14 12 16	0 0 0 0	0 0 0 0	1 5 3 7	16 5 3 10	9 4 15 6
9 13 11 15	0 0 0 0	0 0 0 0	9 13 11 15	9 4 6 15	16 5 10 3
41/P	0/S	0/C	42/P	0/S	0/C
43/P	0/S	0/C	44/P	0/S	0/C
45/P	0/S	0/C	46/P	0/S	0/C
47/P	0/S	0/C	48/P	0/S	0/C
49/P	0/S	25/C	50/P	0/S	26/C
3 1 4 2	0 0 0 0	3 16 2 13	3 1 4 2	0 0 0 0	3 16 2 13
7 5 8 6	0 0 0 0	10 5 11 8	11 9 12 10	0 0 0 0	6 9 7 12
11 9 12 10	0 0 0 0	15 4 14 1	7 5 8 6	0 0 0 0	15 4 14 1
15 13 16 14	0 0 0 0	6 9 7 12	15 13 16 14	0 0 0 0	10 5 11 8
51/P	25/S	27/C	52/P	26/S	28/C
3 1 7 5	3 16 10 5	3 16 5 10	3 1 7 5	3 16 10 5	3 16 5 10
4 2 8 6	13 2 8 11	13 2 11 8	11 9 15 13	6 9 15 4	6 9 4 15
11 9 15 13	6 9 15 4	12 7 14 1	4 2 8 6	13 2 8 11	12 7 14 1
12 10 16 14	12 7 1 14	6 9 4 15	12 10 16 14	12 7 1 14	13 2 11 8
53/P	27/S	29/C	54/P	28/S	30/C
3 1 11 9	3 16 6 9	3 16 9 6	3 1 11 9	3 16 6 9	3 16 9 6
4 2 12 10	13 2 12 7	13 2 7 12	7 5 15 13	10 5 15 4	10 5 4 15
7 5 15 13	10 5 15 4	8 11 14 1	4 2 12 10	13 2 12 7	8 11 14 1
8 6 16 14	8 11 1 14	10 5 4 15	8 6 16 14	8 11 1 14	13 2 7 12
55/P	0/S	31/C	56/P	0/S	32/C
3 4 1 2	0 0 0 0	3 13 2 16	3 4 1 2	0 0 0 0	3 13 2 16
7 8 5 6	0 0 0 0	10 8 11 5	11 12 9 10	0 0 0 0	6 12 7 9
11 12 9 10	0 0 0 0	15 1 14 4	7 8 5 6	0 0 0 0	15 1 14 4
15 16 13 14	0 0 0 0	6 12 7 9	15 16 13 14	0 0 0 0	10 8 11 5
57/P	0/S	0/C	58/P	29/S	33/C
3 4 7 8	0 0 0 0	0 0 0 0	3 4 7 8	3 13 10 8	3 13 8 10
1 2 5 6	0 0 0 0	0 0 0 0	11 12 15 16	6 12 15 1	6 12 1 15
11 12 15 16	0 0 0 0	0 0 0 0	1 2 5 6	16 2 5 11	9 7 14 4
9 10 13 14	0 0 0 0	0 0 0 0	9 10 13 14	9 7 4 14	16 2 11 5

	59/P		0/S		0/C		60/P		30/S		34/C
3	4 11 12	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	3	4 11 12	3 13 6 12	3 13 12 6		
1	2 9 10	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	7	8 15 16	10 8 15 1	10 8 1 15		
7	8 15 16	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1	2 9 10	16 2 9 7	5 11 14 4		
5	6 13 14	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	5	6 13 14	5 11 4 14	16 2 7 9		
	61/P		0/S		0/C		62/P		31/S		0/C
3	7 1 5	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	3	7 1 5	3 10 16 5	0 0 0 0		
4	8 2 6	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	11	15 9 13	6 15 9 4	0 0 0 0		
11	15 9 13	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	4	8 2 6	13 8 2 11	0 0 0 0		
12	16 10 14	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	12	16 10 14	12 1 7 14	0 0 0 0		
	65/P		0/S		0/C		66/P		0/S		0/C
	67/P		0/S		0/C		68/P		0/S		0/C
	69/P		0/S		0/C		70/P		0/S		0/C
	71/P		0/S		0/C		72/P		0/S		0/C
	63/P		0/S		0/C		64/P		32/S		0/C
3	7 4 8	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	3	7 4 8	3 10 13 8	0 0 0 0		
1	5 2 6	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	11	15 12 16	6 15 12 1	0 0 0 0		
11	15 12 16	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1	5 2 6	16 5 2 11	0 0 0 0		
9	13 10 14	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	9	13 10 14	9 4 7 14	0 0 0 0		
	73/P		0/S		35/C		74/P		0/S		36/C
4	2 3 1	0 0 0 0	0 0 0 0	4 15 1 14	4 15 1 14	4	2 3 1	0 0 0 0	0 0 0 0	4 15 1 14	
8	6 7 5	0 0 0 0	0 0 0 0	9 6 12 7	9 6 12 7	12	10 11 9	0 0 0 0	0 0 0 0	5 10 8 11	
12	10 11 9	0 0 0 0	0 0 0 0	16 3 13 2	16 3 13 2	8	6 7 5	0 0 0 0	0 0 0 0	16 3 13 2	
16	14 15 13	0 0 0 0	0 0 0 0	5 10 8 11	5 10 8 11	16	14 15 13	0 0 0 0	0 0 0 0	9 6 12 7	
	75/P		33/S		37/C		76/P		34/S		38/C
4	2 8 6	4 15 9 6	4 15 9 6	4 15 6 9	4 15 6 9	4	2 8 6	4 15 9 6	4 15 9 6	4 15 6 9	
3	1 7 5	14 1 7 12	14 1 7 12	14 1 12 7	14 1 12 7	12	10 16 14	5 10 16 3	5 10 16 3	5 10 3 16	
12	10 16 14	5 10 16 3	5 10 16 3	11 8 13 2	11 8 13 2	3	1 7 5	14 1 7 12	14 1 7 12	11 8 13 2	
11	9 15 13	11 8 2 13	11 8 2 13	5 10 3 16	5 10 3 16	11	9 15 13	11 8 2 13	11 8 2 13	14 1 12 7	
	77/P		35/S		39/C		78/P		36/S		40/C
4	2 12 10	4 15 5 10	4 15 5 10	4 15 10 5	4 15 10 5	4	2 12 10	4 15 5 10	4 15 5 10	4 15 10 5	
3	1 11 9	14 1 11 8	14 1 11 8	14 1 8 11	14 1 8 11	8	6 16 14	9 6 16 3	9 6 16 3	9 6 3 16	
8	6 16 14	9 6 16 3	9 6 16 3	7 12 13 2	7 12 13 2	3	1 11 9	14 1 11 8	14 1 11 8	7 12 13 2	
7	5 15 13	7 12 2 13	7 12 2 13	9 6 3 16	9 6 3 16	7	5 15 13	7 12 2 13	7 12 2 13	14 1 8 11	
	79/P		0/S		41/C		80/P		0/S		42/C
4	3 2 1	0 0 0 0	0 0 0 0	4 14 1 15	4 14 1 15	4	3 2 1	0 0 0 0	0 0 0 0	4 14 1 15	
8	7 6 5	0 0 0 0	0 0 0 0	9 7 12 6	9 7 12 6	12	11 10 9	0 0 0 0	0 0 0 0	5 11 8 10	
12	11 10 9	0 0 0 0	0 0 0 0	16 2 13 3	16 2 13 3	8	7 6 5	0 0 0 0	0 0 0 0	16 2 13 3	
16	15 14 13	0 0 0 0	0 0 0 0	5 11 8 10	5 11 8 10	16	15 14 13	0 0 0 0	0 0 0 0	9 7 12 6	
	81/P		0/S		0/C		82/P		37/S		43/C
4	3 8 7	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	4	3 8 7	4 14 9 7	4 14 9 7	4 14 7 9	
2	1 6 5	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	12	11 16 15	5 11 16 2	5 11 16 2	5 11 2 16	
12	11 16 15	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	2	1 6 5	15 1 6 12	15 1 6 12	10 8 13 3	
10	9 14 13	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	10	9 14 13	10 8 3 13	10 8 3 13	15 1 12 6	
	83/P		0/S		0/C		84/P		38/S		44/C
4	3 12 11	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	4	3 12 11	4 14 5 11	4 14 5 11	4 14 11 5	
2	1 10 9	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	8	7 16 15	9 7 16 2	9 7 16 2	9 7 2 16	
8	7 16 15	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	2	1 10 9	15 1 10 8	15 1 10 8	6 12 13 3	
6	5 14 13	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	6	5 14 13	6 12 3 13	6 12 3 13	15 1 8 10	
	85/P		0/S		0/C		86/P		39/S		0/C
4	8 2 6	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	4	8 2 6	4 9 15 6	4 9 15 6	0 0 0 0	
3	7 1 5	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	12	16 10 14	5 16 10 3	5 16 10 3	0 0 0 0	

12 16 10 14	0 0 0 0	0 0 0 0	3 7 1 5	14 7 1 12	0 0 0 0
11 15 9 13	0 0 0 0	0 0 0 0	11 15 9 13	11 2 8 13	0 0 0 0
	87/P	0/S		88/P	40/S
4 8 3 7	0 0 0 0	0 0 0 0	4 8 3 7	4 9 14 7	0 0 0 0
2 6 1 5	0 0 0 0	0 0 0 0	12 16 11 15	5 16 11 2	0 0 0 0
12 16 11 15	0 0 0 0	0 0 0 0	2 6 1 5	15 6 1 12	0 0 0 0
10 14 9 13	0 0 0 0	0 0 0 0	10 14 9 13	10 3 8 13	0 0 0 0
	89/P	0/S		90/P	0/S
	91/P	0/S		92/P	0/S
	93/P	0/S		94/P	0/S
	95/P	0/S		96/P	0/S
	97/P	0/S		98/P	0/S
5 1 6 2	0 0 0 0	0 0 0 0	5 1 6 2	0 0 0 0	5 16 2 11
7 3 8 4	0 0 0 0	0 0 0 0	13 9 14 10	0 0 0 0	4 9 7 14
13 9 14 10	0 0 0 0	0 0 0 0	7 3 8 4	0 0 0 0	15 6 12 1
15 11 16 12	0 0 0 0	0 0 0 0	15 11 16 12	0 0 0 0	10 3 13 8
	99/P	0/S		100/P	0/S
5 1 7 3	0 0 0 0	0 0 0 0	5 1 7 3	0 0 0 0	5 16 3 10
6 2 8 4	0 0 0 0	0 0 0 0	13 9 15 11	0 0 0 0	4 9 6 15
13 9 15 11	0 0 0 0	0 0 0 0	6 2 8 4	0 0 0 0	14 7 12 1
14 10 16 12	0 0 0 0	0 0 0 0	14 10 16 12	0 0 0 0	11 2 13 8
	101/P	41/S		102/P	42/S
5 1 13 9	5 16 4 9	0 0 0 0	5 1 13 9	5 16 4 9	0 0 0 0
6 2 14 10	11 2 14 7	0 0 0 0	7 3 15 11	10 3 15 6	0 0 0 0
7 3 15 11	10 3 15 6	0 0 0 0	6 2 14 10	11 2 14 7	0 0 0 0
8 4 16 12	8 13 1 12	0 0 0 0	8 4 16 12	8 13 1 12	0 0 0 0
	103/P	0/S		104/P	0/S
	105/P	0/S		106/P	43/S
5 6 7 8	0 0 0 0	0 0 0 0	5 6 7 8	5 11 10 8	0 0 0 0
1 2 3 4	0 0 0 0	0 0 0 0	13 14 15 16	4 14 15 1	0 0 0 0
13 14 15 16	0 0 0 0	0 0 0 0	1 2 3 4	16 2 3 13	0 0 0 0
9 10 11 12	0 0 0 0	0 0 0 0	9 10 11 12	9 7 6 12	0 0 0 0
	107/P	0/S		108/P	0/S
	109/P	0/S		110/P	0/S
	111/P	0/S		112/P	44/S
5 7 6 8	0 0 0 0	0 0 0 0	5 7 6 8	5 10 11 8	0 0 0 0
1 3 2 4	0 0 0 0	0 0 0 0	13 15 14 16	4 15 14 1	0 0 0 0
13 15 14 16	0 0 0 0	0 0 0 0	1 3 2 4	16 3 2 13	0 0 0 0
9 11 10 12	0 0 0 0	0 0 0 0	9 11 10 12	9 6 7 12	0 0 0 0
	113/P	0/S		114/P	0/S
	115/P	0/S		116/P	0/S
	117/P	0/S		118/P	0/S
	119/P	0/S		120/P	0/S
	121/P	0/S		122/P	0/S
6 2 5 1	0 0 0 0	0 0 0 0	6 2 5 1	0 0 0 0	6 15 1 12
8 4 7 3	0 0 0 0	0 0 0 0	14 10 13 9	0 0 0 0	3 10 8 13
14 10 13 9	0 0 0 0	0 0 0 0	8 4 7 3	0 0 0 0	16 5 11 2
16 12 15 11	0 0 0 0	0 0 0 0	16 12 15 11	0 0 0 0	9 4 14 7
	123/P	0/S		124/P	0/S
6 2 8 4	0 0 0 0	0 0 0 0	6 2 8 4	0 0 0 0	6 15 4 9
5 1 7 3	0 0 0 0	0 0 0 0	14 10 16 12	0 0 0 0	3 10 5 16
14 10 16 12	0 0 0 0	0 0 0 0	5 1 7 3	0 0 0 0	13 8 11 2
13 9 15 11	0 0 0 0	0 0 0 0	13 9 15 11	0 0 0 0	12 1 14 7

125/P	45/S	0/C	126/P	46/S	0/C
6 2 14 10	6 15 3 10	0 0 0 0	6 2 14 10	6 15 3 10	0 0 0 0
5 1 13 9	12 1 13 8	0 0 0 0	8 4 16 12	9 4 16 5	0 0 0 0
8 4 16 12	9 4 16 5	0 0 0 0	5 1 13 9	12 1 13 8	0 0 0 0
7 3 15 11	7 14 2 11	0 0 0 0	7 3 15 11	7 14 2 11	0 0 0 0
127/P	0/S	0/C	128/P	0/S	0/C
129/P	0/S	0/C	130/P	47/S	0/C
6 5 8 7	0 0 0 0	0 0 0 0	6 5 8 7	6 12 9 7	0 0 0 0
2 1 4 3	0 0 0 0	0 0 0 0	14 13 16 15	3 13 16 2	0 0 0 0
14 13 16 15	0 0 0 0	0 0 0 0	2 1 4 3	15 1 4 14	0 0 0 0
10 9 12 11	0 0 0 0	0 0 0 0	10 9 12 11	10 8 5 11	0 0 0 0
131/P	0/S	0/C	132/P	0/S	0/C
133/P	0/S	0/C	134/P	0/S	0/C
135/P	0/S	0/C	136/P	48/S	0/C
6 8 5 7	0 0 0 0	0 0 0 0	6 8 5 7	6 9 12 7	0 0 0 0
2 4 1 3	0 0 0 0	0 0 0 0	14 16 13 15	3 16 13 2	0 0 0 0
14 16 13 15	0 0 0 0	0 0 0 0	2 4 1 3	15 4 1 14	0 0 0 0
10 12 9 11	0 0 0 0	0 0 0 0	10 12 9 11	10 5 8 11	0 0 0 0
137/P	0/S	0/C	138/P	0/S	0/C
139/P	0/S	0/C	140/P	0/S	0/C
141/P	0/S	0/C	142/P	0/S	0/C
143/P	0/S	0/C	144/P	0/S	0/C
145/P	0/S	0/C	146/P	0/S	0/C
147/P	0/S	0/C	148/P	0/S	0/C
149/P	0/S	0/C	150/P	0/S	0/C

[Counts = 384/P, 48/S, 48/C] OK!

#5. あらためて「原初方陣」とは何か

正直に告白するが、原初方陣とひな形変換法の元々のアイデアは、小生のオリジナルな発明ではない。元ネタは、英国で活躍しているオルレンショー&ブラエ両氏の提唱だ。四次・八次・十二次・十六次の専ら「正方連結型完全方陣」について解の総数計算法を発見したという報告の中に出て来た方法だ。小生は、その概要を鈴木教授に教わった。

いろいろ自分で調べてみて、これは正方連結陣に限らず、どの次数・どのタイプの魔方陣でも成り立つ概念ではないかと、私は直感した。つまり、原初方陣およびひな形変換法は、魔方陣全般に普遍的に見られる現象であり、しかも極めて基礎的な方法であると確信した。

それを何とかわかりやすく取り出すことはできないか。低次から始めていろいろ実験して調べた事柄を、今現在こうして各論的に報告しているわけなのである。

「原初方陣 ('Prototype Squares')」および「ひな形変換法 ('Do-it-After-the-Model Transformation')」という名前は、私が自分で付けた。元々の概念をだいぶ拡張・修正しなければならず、ほとんど別のものに作り変えてしまったからである。

オルレンショー&ブラエ両氏は、論理的必然性を強調する余り、原初方陣もひな形変換法も、ただ1通りしか正しいものは無いように言っているが、これは違う。事實は、必ずしもそうっていない。最良のひな形は、決して多数ではないが、複数個見つかる。次数・タイプによっては、全ての標準解がひな形になれるものもある。目的方陣を構成できるのだ。

原初方陣の構成条件も、必ずしも1通りに限らない。強調されている「クロスサムの等和」なども、必ずしも必要不可欠の必須条件ではない。いや、無くても全然かまわない。

正方連結陣の場合のように、あると便利な事例もあるかわりに、奇数次の或るタイプの魔方陣のように、「クロスサムの等和」条件があると、困る事例さえあるのである。

小生の昔の原稿では、この「クロスサムの等和」条件を詳しく解説したものだが、それは「勉強」の成果を記したまで。今はもう全然説明しなくてよいと考えている。

それにしても、原初方陣というのは一体何ものなのだろう。あらためて不思議に思う。

最良のひな形は複数個可能なのに、原初方陣の解集合は、その次数・その型に1通りしか無い。しかも、これが最も不思議なのだが、必ず有る。目的方陣と同じ総数の解を持った原初方陣の集合が、1個ではあるが、必ず見つかるのである。

四次の場合は特に、補数対称型も、汎対角線型も、完全型も、正方連結型汎魔方陣の場合も、原初方陣の解集合は1つしかなく、全部に共通なのだ。これは、何を意味するのか。

原初方陣を「基本図の集合」と考えると、四次では「基本になる方陣」の種類が本質的に同じで、1種類しか無いという意味になる。現実の多様な魔方陣は、その基本方陣の「派生形」、あるいは個性的な「現象」ということになる。

何だか考え難いことではあるが、そう推理せざるを得ない。

#6. 「プリンシパル方陣」と大基本解

ところで、四次の場合は、対称方陣でも汎魔方陣でも大基本解が3個存在する。

その3個があれば、後は簡単な変換法で作ることができる。3×16=48個の標準解と、それを反転・回転して得られる48×8=384個の原始解を、その大基本解3個から作ることができる。

では、その大基本解3個に対応する原初四方陣は、どんな形をしているのか。

はたして、取り立てて言うほどの特徴があるのか。

* 原初四方陣と対称&汎四方陣の三大基本形 *

1/PT	/SC	/PD
1 2 3 4	1 15 14 4	1 15 4 14
5 6 7 8	12 6 7 9	12 6 9 7
9 10 11 12	8 10 11 5	13 3 16 2
13 14 15 16	13 3 2 16	8 10 5 11
3/PT	/SC	/PD
1 2 5 6	1 15 12 6	1 15 6 12
3 4 7 8	14 4 7 9	14 4 9 7
9 10 13 14	8 10 13 3	11 5 16 2
11 12 15 16	11 5 2 16	8 10 3 13
5/PT	/SC	/PD
1 2 9 10	1 15 8 10	1 15 10 8
3 4 11 12	14 4 11 5	14 4 5 11
5 6 13 14	12 6 13 3	7 9 16 2
7 8 15 16	7 9 2 16	12 6 3 13

大基本形3個を得るには、対称四方陣の場合は、標準解の中から $n1=1; n2>n3; n5>n9;$ の条件を満たすものを選び出せば良い。汎四方陣の場合は、標準解の中から $n1=1; n2>n4; n5>n13;$ の条件を満たすものだけを選び出せば良い。不等式の向きは逆でも構わない。

では、その3形を産み出す原初方陣には、共通するどんな特徴があるか。

それは、 $(n1<n2<n3<n4) \&\& (n1<n5<n9<n13) \&\& (n1<n16) \&\& (n2<n5)$ だ。

ただし、不等式の向きは、このまま。逆に書いてはいけない。

この美しい成立条件を発見したのは、前述のオルレンショー&ブラエ両氏だ。彼らは「プリンシパル方陣」と名付けて大基本解扱いをしている。そして、これを原初方陣の構造解析に結び付けて、正方連結陣の解の総数計算法の根拠としている。

この条件と大基本解とが一对一对応して成り立つのは、この表の場合だけだろうか。ざっと見た感じでは、他のひな形を用いた場合にはそうならない。

しかし、何事も詳しく調べなければわからない。試みに大基本解の1つ1つを「ひな形」にして各個変換してみよう。用いる原初四方陣は、当然「プリンシパル方陣」である。

* 『プリンシパル方陣』と完全四方陣の大基本解3個の対応関係 *

(完全四方陣の大基本解の方は $\{n1=1; n2>n4; n5>n13; \text{ and } n2>n5;\}$ で求めた)

1/P	1/C	2/P	2/C	3/P	3/C
1 2 3 4	1 15 4 14	1 2 5 6	1 15 6 12	1 2 9 10	1 15 10 8
5 6 7 8	12 6 9 7	3 4 7 8	14 4 9 7	3 4 11 12	14 4 5 11
9 10 11 12	13 3 16 2	9 10 13 14	11 5 16 2	5 6 13 14	7 9 16 2
13 14 15 16	8 10 5 11	11 12 15 16	8 10 3 13	7 8 15 16	12 6 3 13
1/P	2/C	2/P	1/C	3/P	0/C
1 2 3 4	1 15 6 12	1 2 5 6	1 15 4 14	1 2 9 10	1 15 4 14
5 6 7 8	14 4 9 7	3 4 7 8	12 6 9 7	3 4 11 12	8 10 5 11
9 10 11 12	11 5 16 2	9 10 13 14	13 3 16 2	5 6 13 14	13 3 16 2
13 14 15 16	8 10 3 13	11 12 15 16	8 10 5 11	7 8 15 16	12 6 9 7
1/P	3/C	2/P	0/C	3/P	0/C
1 2 3 4	1 15 10 8	1 2 5 6	1 15 10 8	1 2 9 10	1 15 6 12
5 6 7 8	14 4 5 11	3 4 7 8	12 6 3 13	3 4 11 12	8 10 3 13
9 10 11 12	7 9 16 2	9 10 13 14	7 9 16 2	5 6 13 14	11 5 16 2
13 14 15 16	12 6 3 13	11 12 15 16	14 4 5 11	7 8 15 16	14 4 9 7

[Count = 3/3]

第1番解は、他の2解を再現している。だが、第2、3番解は、他の2解を再現できていない。ひな形としては、第1番解しか使えないことがわかる。

他には無いか。試みに、大基本解の作り方を変えて、別の3個セットを用意しよう。別の解に見えるが、内容的・本質的に同じものだ。整理の仕方が違うだけである。

*** 『プリンシパル方陣』と完全四方陣の大基本解3個の対応関係 ***

(完全四方陣の大基本解の方は {n1==1; n2<n4; n5<n13; and n4<n13;} で求めた)

1/P	1/C	2/P	2/C	3/P	3/C
1 2 3 4	1 8 13 12	1 2 5 6	1 8 11 14	1 2 9 10	1 12 7 14
5 6 7 8	14 11 2 7	3 4 7 8	12 13 2 7	3 4 11 12	8 13 2 11
9 10 11 12	4 5 16 9	9 10 13 14	6 3 16 9	5 6 13 14	10 3 16 5
13 14 15 16	15 10 3 6	11 12 15 16	15 10 5 4	7 8 15 16	15 6 9 4
1/P	2/C	2/P	1/C	3/P	0/C
1 2 3 4	1 8 11 14	1 2 5 6	1 8 13 12	1 2 9 10	1 12 13 8
5 6 7 8	12 13 2 7	3 4 7 8	14 11 2 7	3 4 11 12	14 7 2 11
9 10 11 12	6 3 16 9	9 10 13 14	4 5 16 9	5 6 13 14	4 9 16 5
13 14 15 16	15 10 5 4	11 12 15 16	15 10 3 6	7 8 15 16	15 6 3 10
1/P	3/C	2/P	0/C	3/P	0/C
1 2 3 4	1 12 7 14	1 2 5 6	1 14 7 12	1 2 9 10	1 14 11 8
5 6 7 8	8 13 2 11	3 4 7 8	8 11 2 13	3 4 11 12	12 7 2 13
9 10 11 12	10 3 16 5	9 10 13 14	10 5 16 3	5 6 13 14	6 9 16 3
13 14 15 16	15 6 9 4	11 12 15 16	15 4 9 6	7 8 15 16	15 4 5 10

[Count = 3/3]

この場合も、第1番解は他の2解を再現している。だが、他の2解は、それぞれその他を再現できていない。つまり、第1番解しか「ひな形」に使えないことがわかる。

しかし、この第1番解は、前の第1番解とは明らかに外見が違う。これを別個と数えるならば、ひな形候補には少なくとも2個があがることになる。この際「プリンシパル方陣」の方は一切変えていないから、対応するひな形は、ただ1個だけではないと言える。

しかし、この2個は同時には成立しない。あくまで整理の仕方を変えたために出て来たのであって、1つの整理の仕方を取る限り、どちらか一方しか出て来ない基本解である。その意味では、プリンシパル方陣に対応する「ひな形」は本質的に1個しか無いとも言える。

この辺は、厳密に議論し、自分なりの選択を明確に宣言しておかなければならない。

我々は、えてして強調表現の多い独断論に説得されがちだが、あくまで「事実」或いは「真理」に従うべきであると、小生は考える。懐疑こそが、科学の原動力だからだ。

予想が外れ、理論が破綻しているなら、たとえ相手がオイラー先生のような高名な権威でも、それを敢然と指摘しなければならない。そして、もし自分が指摘された場合には、それを素直に認める勇気を持たなければならないと、自戒している。

(On Mar. 3, 2001; Mar. 8, 2003; Feb. 27, 2006; 最新改訂稿 : Aug. 2, 2010;
MacOSX & Xcode3.1.2 による作表・計数・執筆 : 摂田 寛二(Kanji Setsuda))
E-Mail Address: jag12001@nifty.com

[附録 : プログラム・リスト]

```
/** Comparison of 'Prototype' Squares of Order 4 between **/  
/** Self-Complementary Type and Complete Pan-magic Type **/  
/** 'ProtoT4S&C384.c' Dictated by Kanji Setsuda **/  
/** on Mar.8, 2003; Sep.26, 2006; Aug.1, 2010 **/  
/** Working with MacOSX & Xcode 3.1.2 **/  
/**/  
#include <stdio.h>  
/**/  
short cnt1, cnt2, cnt3, tcnt;  
short LSM, PSM;  
short nm[17], uflg[17], dm[17];  
short d[7][17];  
short pts[385][17];  
short ptc[385][17];  
/**/  
/* Sub-Routines */  
void sstp01(void), sstp02(void), sstp03(void);  
void sstp04(void), sstp05(void), sstp06(void);  
void sstp07(void), sstp08(void), sstp09(void);  
void recspspr(void), pr6ans(void);  
void cstp01(void), cstp02(void), cstp03(void);  
void cstp04(void), cstp05(void), cstp06(void);  
void cstp07(void), cstp08(void), cstp09(void);  
void reccpspr(void);  
void pr3pairs(short x);  
void damts(void), damtc(void);  
void prp2trios(short x);  
void prs2trios();  
/**/  
/* Main Program */  
int main(){  
short m,n;  
printf("\n** Comparison of 'Prototype' Squares of Order 4 between **\n");  
printf("** Self-Complementary Type and Complete Pan-magic Type **\n");  
printf("\n** 'Prototype' Squares of Order 4 for Self-complementary Type **\n");  
PSM=17; LSM=34;  
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}  
cnt1=0; cnt3=0;  
sstp01(); //Make 'Prototype Squares' for Self-complementary Type */  
printf(" [Count = %d]\n",cnt1);  
printf("\n** 'Prototype' Squares of Order 4 for Complete Pan-magic Type **\n");  
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}  
cnt2=0; cnt3=0;  
cstp01(); //Make 'Prototype Squares' for Complete Pan-magic Type */  
printf(" [Count = %d]\n",cnt2);  
printf("\n** Compare 'Prototype' Squares 4x4 between S-C and Complete PM Types **\n");  
pr3pairs(50);  
if(m<cnt1){printf(" . . . . \n");}
```

```

printf(" [Counts = %d/S : %d/C]\n",cnt1,cnt2);
tcnt=cnt1; cnt1=0; cnt2=0;
printf("\n *** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ***\n");
printf(" * Primitive Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C *\n");
prp2trios(50);
printf(" . . . . .\n");
printf(" [Counts = %d/P, %d/S, %d/C]\n",tcnt,tcnt,tcnt);
cnt1=0; cnt2=0;
printf("\n *** Two Types of Magic Squares 4x4 Recomposed by 'DAMT' from Prototypes: ***\n");
printf(" * Standard Sets: Prototype/P, Self-complementary/S, Complete Pan-magic/C *\n");
prs2trios();
printf(" . . . . .\n");
printf(" [Counts = %d/P, %d/S, %d/C] OK!\n",tcnt,cnt1,cnt2);
return 0;
}
/**/
/* Make 'Prototype Squares' of Order 4 and Apply 'DAMT' */
/** Basic Forms for Prototype Squares for Self-complementary Type **
 3 2 16 13 3 2 16 13 3 2
15 14 4 | 1|15|14| 4 | 1 15 14      | 1 | 2 | 3 | 4 |
6 7 9 |12| 6 | 7 | 9 |12 6 7      | 5 | 6 | 7 | 8 |
10 11 5 | 8|10|11| 5 | 8 10 11      | 9 |10|11|12|
3 2 16 |13| 3 | 2 |16 |13 3 2      |13|14|15|16|
15 14 4 1 15 14 4 1 15 14
** Basic Conditions for Prototype Squares 4x4: **
n1+n15+n14+n4=C ...rw1; | n1+n12+n8+n13=C ...c1;
n12+n6+n7+n9=C ...rw2; | n15+n6+n10+n3=C ...c2;
n8+n10+n11+n5=C ...rw3; | n14+n7+n11+n2=C ...c3;
n13+n3+n2+n16=C ...rw4; | n4+n9+n5+n16=C ...c4;
** Self-complementary Conditions for Prototype Squares 4x4: **
n1+n16=n15+n2=n14+n3=n4+n13=n12+n5=n6+n11=n7+n10=n9+n8=CC ...scc
*/
/* Set N1 & n16=PSM-N1 */
void sstp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
sstp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N15 & n2=PSM-N15 */
void sstp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
sstp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N14 & n3=PSM-N14 */
void sstp03(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){

```

```

        nm[14]=a; nm[3]=b;
        uflg[a]=1; uflg[b]=1;
        sstp04();
        uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n4=LSM-n1-n15-n14 & n13 */
void sstp04(){
    short a,b;
    a=LSM-nm[1]-nm[15]-nm[14];
    if((0<a)&&(a<17)){
        b=LSM-nm[3]-nm[2]-nm[16];
        if(a+b==PSM){
            if((uflg[a]==0)&&(uflg[b]==0)){
                uflg[a]=1; uflg[b]=1;
                nm[4]=a; nm[13]=b;
                sstp05();
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set N12 & n5=PSM-N12 */
void sstp05(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[12]=a; nm[5]=b;
            sstp06();
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set N6 & n11=PSM-n6 */
void sstp06(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            uflg[a]=1; uflg[b]=1;
            nm[6]=a; nm[11]=b;
            if(nm[1]+nm[6]+nm[11]+nm[16]==LSM){sstp07();}
            uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void sstp07(){
    short a,b;
    a=LSM-nm[1]-nm[12]-nm[13];
    if((0<a)&&(a<17)){
        b=LSM-nm[16]-nm[5]-nm[4];
        if(a+b==PSM){
            if((uflg[a]==0)&&(uflg[b]==0)){
                uflg[a]=1; uflg[b]=1;
                nm[8]=a; nm[9]=b;
                sstp08();
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void sstp08(){
    short a,b,c,d;
    a=LSM-nm[12]-nm[6]-nm[9];
    if((0<a)&&(a<17)){
        b=LSM-nm[8]-nm[11]-nm[5];
        if(a+b==PSM){
            c=LSM-nm[14]-nm[11]-nm[2];
            d=LSM-nm[15]-nm[6]-nm[3];
            if((a==c)&&(b==d)){

```

```

        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[7]=a; nm[10]=b;
            uflg[a]=1; uflg[b]=1;
            if(nm[4]+nm[7]+nm[10]+nm[13]==LSM){recpspr();}
            uflg[b]=0; uflg[a]=0;}}}}
    }
/**/
/* Record the Solutions and Print */
void recpspr(){
    short n;
    for(n=1;n<17;n++){d[cnt3][n]=nm[n]; pts[cnt1][n]=nm[n];}
    pts[cnt1][0]=cnt1+1;
    cnt1++; d[cnt3][0]=cnt1;
//cnt3++; if(cnt3==6){pr6ans(); cnt3=0;}
}
/**/
/* Print 3 Pairs of Solutions */
void pr6ans(){
    short l,l4,m,n;
    printf("%11d/%13d/%13d/%13d/%13d/%13d/\n",
        d[0][0],d[1][0],d[2][0],d[3][0],d[4][0],d[5][0]);
    for(l=0;l<4;l++){l4=l*4;
        for(m=0;m<6;m++){
            for(n=1;n<5;n++){printf("%3d",d[m][l4+n]);}
            if(m<5){printf(" ");}
        }
        printf("\n");
    }
}
/**/
/* Make 'Prototype Squares' of Order 4 and Apply 'DAMT' */
/** Basic Forms for Prototype Squares for Complete Pan-magic Type */
/** Basic Conditions for Prototype Squares 4x4: **
    n1+n15+n4+n14=C ...rw1; | n1+n12+n13+n8=C ...c1;
    n12+n6+n9+n7=C ...rw2; | n15+n6+n3+n10=C ...c2;
    n13+n3+n16+n2=C ...rw3; | n4+n9+n16+n5=C ...c3;
    n8+n10+n5+n11=C ...rw4; | n14+n7+n2+n11=C ...c4;

10 5 11 8 10 5 11 8 10 5
15 4 14 | 1|15| 4|14| 1 15 4 | 1| 2| 3| 4|
6 9 7 |12| 6| 9| 7|12 6 9 | 5| 6| 7| 8|
3 16 2 |13| 3|16| 2|13 3 16 | 9|10|11|12|
10 5 11| 8|10| 5|11| 8 10 5 |13|14|15|16|
15 4 14 1 15 4 14 1 15 4

** Complementary Conditions for Prototype Squares 4x4: **
n1+n16=n15+n2=n4+n13=n14+n3=n12+n5=n6+n1=n9+n81=n7+n10=CC ...cc
*/
/* Set N1 & n16=PSM-N1 */
void cstp01(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[1]=a; nm[16]=b;
            uflg[a]=1; uflg[b]=1;
            cstp02();
            uflg[b]=0; uflg[a]=0;}
        }
}
}
/* Set N15 & n2=PSM-N15 */

```

```

void cstp02(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[15]=a; nm[2]=b;
uflg[a]=1; uflg[b]=1;
cstp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N4 & n13=PSM-N4 */
void cstp03(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[4]=a; nm[13]=b;
cstp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n14=LSM-n1-n15-n4 & n3 */
void cstp04(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[4];
if((0<a)&&(a<17)){
b=LSM-nm[16]-nm[2]-nm[13];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[14]=a; nm[3]=b;
cstp05();
uflg[b]=0; uflg[a]=0;}}}
}
}
/* Set N12 & n5=PSM-N12 */
void cstp05(){
short a,b;
for(a=16;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[12]=a; nm[5]=b;
uflg[a]=1; uflg[b]=1;
cstp06();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set N6 & n11=PSM-N6 */
void cstp06(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
uflg[a]=1; uflg[b]=1;
nm[6]=a; nm[11]=b;
if(nm[1]+nm[6]+nm[16]+nm[11]==LSM){cstp07();}
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n8=LSM-n1-n12-n13 & n9 */
void cstp07(){
short a,b;
a=LSM-nm[1]-nm[12]-nm[13];
if((0<a)&&(a<17)){
b=LSM-nm[16]-nm[5]-nm[4];
if(a+b==PSM){
if((uflg[a]==0)&&(uflg[b]==0)){

```

```

    uflg[a]=1; uflg[b]=1;
    nm[8]=a; nm[9]=b;
    if(nm[14]+nm[9]+nm[3]+nm[8]==LSM){cstp08();}
    uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n12-n6-n9 & n10 */
void cstp08(){
short a,b,c,d;
a=LSM-nm[12]-nm[6]-nm[9];
if((0<a)&&(a<17)){
b=LSM-nm[5]-nm[11]-nm[8];
if(a+b==PSM){
c=LSM-nm[14]-nm[2]-nm[11];
d=LSM-nm[3]-nm[15]-nm[6];
if((a==c)&&(b==d)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[7]=a; nm[10]=b;
uflg[a]=1; uflg[b]=1;
reccpspr();
uflg[b]=0; uflg[a]=0;}}}}
}
/**/
/* Print the Solutions */
void reccpspr(){
short n;
for(n=1;n<17;n++){d[cnt3][n]=nm[n]; ptc[cnt2][n]=nm[n];}
ptc[cnt2][0]=cnt2+1;
cnt2++; d[cnt3][0]=cnt2;
//cnt3++; if(cnt3==6){pr6ans(); cnt3=0;}
}
/**/
/* Print 3 Pairs of Solutions */
void pr3pairs(short x){
short l,l4,m,n,p;
for(m=0;m<x;m=m+3){
printf("%10d/S%4d/C%19d/S%4d/C%19d/S%4d/C\n",
pts[m][0],ptc[m][0],pts[m+1][0],ptc[m+1][0],pts[m+2][0],ptc[m+2][0]);
for(l=0;l<4;l++){l4=l*4;
for(n=m;n<(m+3);n++){
for(p=1;p<5;p++){printf("%3d",pts[n][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",ptc[n][l4+p]);}
if(n<(m+2)){printf(" ");}
}
printf("\n");
}
}
}
/**/
/* Do-it-After-the-Model Transformation System */
/* ProtoT/ ModelS/ ModelC/


|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |



|    |    |    |    |
|----|----|----|----|
| 1  | 15 | 14 | 4  |
| 12 | 6  | 7  | 9  |
| 8  | 10 | 11 | 5  |
| 13 | 3  | 2  | 16 |



|    |    |    |    |
|----|----|----|----|
| 1  | 15 | 4  | 14 |
| 12 | 6  | 9  | 7  |
| 13 | 3  | 16 | 2  |
| 8  | 10 | 5  | 11 |


*/
/* Transform Prototype into Self-Complementary Object */
void damts(void){
dm[1]=nm[1]; dm[2]=nm[15]; dm[3]=nm[14]; dm[4]=nm[4];

```

```

dm[5]=nm[12]; dm[6]=nm[6]; dm[7]=nm[7]; dm[8]=nm[9];
dm[9]=nm[8]; dm[10]=nm[10]; dm[11]=nm[11]; dm[12]=nm[5];
dm[13]=nm[13]; dm[14]=nm[3]; dm[15]=nm[2]; dm[16]=nm[16];
}
/**/
/* Transform Prototype into Complete Pan-magic Object */
void damtc(void){
dm[1]=nm[1]; dm[2]=nm[15]; dm[3]=nm[4]; dm[4]=nm[14];
dm[5]=nm[12]; dm[6]=nm[6]; dm[7]=nm[9]; dm[8]=nm[7];
dm[9]=nm[13]; dm[10]=nm[3]; dm[11]=nm[16]; dm[12]=nm[2];
dm[13]=nm[8]; dm[14]=nm[10]; dm[15]=nm[5]; dm[16]=nm[11];
}
/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Primitive Set] */
void prp2trios(short x){
short l,l4,m,n,p;
for(m=0;m<x;m=m+2){
for(n=1;n<17;n++){nm[n]=pts[m][n];}
damts(); for(n=1;n<17;n++){d[0][n]=dm[n];}
for(n=1;n<17;n++){nm[n]=ptc[m][n];}
damtc(); for(n=1;n<17;n++){d[1][n]=dm[n];}
for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
damts(); for(n=1;n<17;n++){d[2][n]=dm[n];}
for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
damtc(); for(n=1;n<17;n++){d[3][n]=dm[n];}
printf("%10d/P /S /C%12d/P /S /C\n",
pts[m][0],pts[m+1][0]);
for(l=0;l<4;l++){l4=l*4;
for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
printf(" ");
for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
printf("\n");
}
}
}
/**/
/* Print 2 Trios of Recomposed Solutions by DAMT [Standard Set] */
void prs2trios(){
short l,l4,m,n,p;
for(m=0;m<tcnt;m=m+2){
for(n=0;n<4;n++){for(p=0;p<17;p++){d[n][p]=0;}}
for(n=1;n<17;n++){nm[n]=pts[m][n];}
damts();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
cnt1++; d[0][0]=cnt1; for(n=1;n<17;n++){d[0][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=ptc[m][n];}
damtc();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
cnt2++; d[1][0]=cnt2; for(n=1;n<17;n++){d[1][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=pts[m+1][n];}
damts();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){
cnt1++; d[2][0]=cnt1; for(n=1;n<17;n++){d[2][n]=dm[n];}}
for(n=1;n<17;n++){nm[n]=ptc[m+1][n];}
damtc();
if((dm[1]<dm[4])&&(dm[1]<dm[13])&&(dm[1]<dm[16])&&(dm[5]<dm[2])){

```



```

    cnt2++; d[3][0]=cnt2; for(n=1;n<17;n++){d[3][n]=dm[n];}}
printf("%10d/P%11d/S%11d/C%12d/P%11d/S%11d/C\n",
pts[m][0],d[0][0],d[1][0],pts[m+1][0],d[2][0],d[3][0]);
if((d[0][0]>0)||(d[1][0]>0)||(d[2][0]>0)||(d[3][0]>0)){
    for(l=0;l<4;l++){l4=l*4;
        for(p=1;p<5;p++){printf("%3d",pts[m][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[0][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",pts[m+1][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[2][l4+p]);}
        printf(" ");
        for(p=1;p<5;p++){printf("%3d",d[3][l4+p]);}
        printf("\n");
    }
}
}
}
/**/

```